Araştırma Yazılımları için Bilimsel Altyapılar

DANIŞMA KAPALI
İstişare dönemi 21 Ekim'den 10 Kasım'a kadar sürdü. **İstişare şimdi kapandı.** Araştırma Yazılımları için Bilimsel Altyapılar hakkındaki raporun son versiyonuna doğru ilerlerken, alınan tüm yorumlar dikkate alınacaktır.

Şuradan rapor:
EOSC Yönetim Kurulu
Çalışma Grubu (ÇG) Mimarisi
Görev Gücü (TF) SIRS

İlk taslak
Ekim 2020

EOSC Architecture
Executive Board Working Group

"Programlama, bir başka insana bilgisayarın yapmasını istediğini söyleme sanatıdır." - Donald Knuth

Önsöz

Yazılım, insanların bilgiyi paylaşması için başka bir araç haline geldi. Araştırma durumunda, yazılım, programlama dillerini kullanarak bilgiyi sağlar, aynı şekilde yayınlar da doğal dilleri kullanarak bilgi verir.

21. yüzyılda, birçok araştırma etkinliği, deneylerini izlemek, sonuçlarını görselleştirmek veya analiz etmek veya simülasyon yoluyla hipotezleri kontrol etmek için bilgisayar sistemlerini kullanır.

Bu nedenle araştırma yazılımlarının arşivlenmesi, korunması ve paylaşılması zorunlu hale gelmiştir. Yayınların ve verilerin yanında yazılımların kalıcılığını ve kullanılabilirliğini sağlamak için öncü çalışmalar başlamıştır.

Avrupa'yı Açık Bilime yönlendirme sözünü yerine getirmek için, Avrupa Açık Bilim Bulutu programı olan EOSC, Araştırma Yazılımları için Bilimsel Altyapıların geliştirilmesini desteklemek zorundadır. Haziran 2020'de, EOSC Mimari Çalışma Grubu (WG), bu konuyla ilgili öneriler sunma yetkisine sahip bir Görev Gücü (TF) başlattı. Araştırma yazılımı altyapılarına dahil olan dokuz kuruluşun ortaklığıyla, TF bu raporu dört ay içinde hazırlayabildi.

Son teknoloji araştırma yazılımı altyapılarının ilk görünümünü sağladıktan sonra, rapor en iyi uygulamaları önerir, açık sorunları tanımlar ve kullanım örneklerini açıklar. Bu temelde, hem teknik hem de politika seviyelerinde, hem kısa vadeli hem de uzun vadeli ufuklar ile tavsiyeler sunulmaktadır.

EOSC Architecture WG'nin başkanı olarak, çalışmanın derinliği ve raporun sunduğu geleceğe bakış için TF üyelerine içtenlikle teşekkür etmek istiyorum.

Jean-François Abramatik

16 Ekim 2020

1. Yönetici Özeti

mevcut operasyonel altyapıların kapsamlarını ve yaklaşımlarını karşılaştıran temsili bir panelin incelenmesi ve belgelenmesi üzerine kurulmuştur.

Bu rapor, en son durumu özetler, en iyi uygulamaları ve açık sorunları tanımlar ve EOSC'nin yazılım ayağını desteklemek açısından farklı yaklaşımları birleştirmenin yolunu açar.

Yeniliğin yakıtı, endüstrilerimizin motoru ve akademik araştırmanın temel direği olan yazılım, modern bilimsel araştırmanın gerekli bir bileşenidir. Bu nedenle, yazılım geliştirmeleri birçok alan ve disiplinde ortaya çıkmaktadır. Ne yazık ki, yazılımın aslında insanlar tarafından insanlar tarafından okunmak üzere tasarlanmış, makineler tarafından *yazılım kaynak kodu biçiminde yürütülen özel bir bilgi biçimi olduğu önemli bir gerçektir.*. Yazılım kaynak kodu, veri görselleştirme, veri analizi, veri dönüştürme ve genel olarak veri işlemenin bilimsel makalelerde elde edilebileceklerin çok ötesine geçen bir kesinlik düzeyiyle tanımlanmasına olanak tanır. Araştırma projelerinde kullanılan yazılımlara erişim olmadan bilimsel sonuçları yeniden üretmenin ve diğer araştırmacılar tarafından elde edilen sonuçların üzerine inşa etmenin son derece zor olduğu artık iyi bilinmektedir.

Son on yılda, bilimsel dünyada yazılımın önemi konusunda farkındalık arttı. Yazılımı bilimsel dünyada birinci sınıf bir vatandaş yapmak için üstesinden gelinmesi gereken dört temel zorluktan bazılarını ele almak için çeşitli altyapılar inşa edilmeye veya uyarlanmaya başlandı :

1. Araştırma yazılımı eserlerinin kaybolmamasını sağlamak için *arşivleme yazılımı.*
2. Araştırma eserlerinin tam olarak tanımlanabilmesini sağlamak için *referans yazılımı.*
3. Araştırma yazılımı eserlerini kolayca keşfetmek ve tanımlamak için yazılımı tanımlama .
4. Katkılarının tanınmasını sağlamak için tüm yazarlara kredi vermek .

Bu zorlukları ele almaya başlamak için, TF , Avrupa çapındaki mevcut operasyonel altyapılardan temsilcilerle birlikte EOSC Mimarisi Çalışma Grubu temsilcileri tarafından oluşturulmuştur (Bölüm 2.2'de sunulmuştur). TF , *arşivlerin* , *yayıncıların* ve *toplayıcıların (kataloglar dahil)* tüm yelpazesini kapsar ve dört sütunun oluşturulmasıyla ilgili bazı zorlukları ele alma konusundaki geniş kapsamlı deneyimlerine dayalı olarak temsili bir panel olarak kabul edilir .

TF , bu ihtiyaçların ele alınmasının standartlar *oluşturmayı* , *araçlar geliştirmeyi, altyapıları iyileştirmeyi ve birbirine bağlamayı, eğitimi, erişimi* ve *yayıncılık topluluğuyla katılımı gerektireceğini düşünmektedir* . *Hem geliştirme, iletişim* ve *erişim çabaları* hem de *operasyonel maliyetler için uygun finansman sağlanması* gerekecektir .

TF, mevcut ihtiyaçların ve son teknolojinin analizinden ve geleceğin mimarisinin tasarımından ortaya çıkan bir dizi öneriyi somut olarak sundu. Bunlar, kısa vadeli eyleme geçirilebilir öğeleri, EOSC için daha geniş politika önerilerini ve ayrıca daha uzun vadeli bir perspektifi içerir.

Kısa vadeli önerilerin 2 – 4 yıllık bir zaman diliminde somut kalkınma projelerine dönüştürülmesi öngörülmektedir . Bu raporun sonunda ayrıntıları verilen somut tavsiyeler, (i) arşivler, yayıncılar ve toplayıcılar arasındaki *etkileşimleri güçlendirmeyi* , *(ii) meta veri standartlarını benimsemeyi* , (iii) *yazılım için dışsal ve içsel tanımlayıcıların kullanımını genelleştirmeyi* , (iv) amaçlarına sahiptir.) araştırma yazılımı kaynak kodu için *uygun alıntıları sağlamak* , *(v) politika* ve *yönergeler yoluyla standardizasyonu teşvik etmek* ve (vi) *araştırma* topluluğu için süreçlerin ve araçların kolayca *benimsenmesini sağlamak.*

TF, genel mimarinin araştırma topluluğunun ihtiyaçlarını en iyi şekilde karşılayacak şekilde inşa edilmesini sağlamada EOSC'nin oynayacağı kilit bir role sahip olduğunu öngörmektedir. Açıklığı, şeffaflığı ve iyi yönetişimi sağlamak için EOSC, katılımcı altyapıları için bu ilkeleri içeren bir dizi mükemmellik kriteri hazırlamalı ve somut tavsiyelerde bulunmalıdır. Ek olarak, EOSC, yazılım için temel altyapılara aktif olarak dahil olmalı, stratejik evriminde yer almalı ve uzun vadeli sürdürülebilirliklerini sağlamak için uygun finansman sağlamalıdır.

önemlidir. Ayrıca, makaleler, veriler ve yazılımlar gibi farklı araştırma çıktıları arasında uygun bir entegrasyonu ele almak için teknoloji ve araçlar araştırılmalıdır.

Son olarak, TF, araştırma yazılımının varsayılan olarak bir Açık Kaynak lisansı altında sunulmasını talep etmek için gelecekteki tüm araştırma finansman programlarına bir madde eklenmesini ve bu varsayılandan tüm sapmaların usulüne uygun olarak motive edilmesi gerektiğini şiddetle tavsiye eder. EOSC, tüm araştırma çıktılarının *"mümkün olduğunca açık, gerektiği kadar kapalı" olması gerektiği şeklindeki genel ifadeye katılıyor olsa* da, yazılımın birinci sınıf vatandaş olarak ele alınması için bu varsayılanın uyarılması gerektiğine inanılıyor.

Rapor 26 Ekim'den 10 Kasım'a kadar Açık Danışmaya tabi tutulacaktır.

2. Giriş

Yazılım, modern bilimsel ekosistemde temel bir bileşen haline geldi ve yazılım geliştirmeleri tüm alanlarda ve disiplinlerde ortaya çıkıyor (Van Noorden ve diğerleri, 2014) . Araştırma projelerinde kullanılan yazılıma erişim olmadan, verilerin yeniden kullanım (Baker, 2016a) ve doğrulama (Barnes, 2010; Stodden ve diğerleri, 2012) için daha az uygun olduğu artık iyi bilinmektedir .

Bilimsel dünyada, yazılım genellikle sadece bir araç olarak görülmüş, yazılımın aslında özel bir bilgi biçimi olduğu, insanlar tarafından insanlar için *yazılım kaynak kodu biçiminde yazılmış* ve ancak daha sonra çalıştırılabilir koda dönüştüğü önemli gerçeği göz ardı edilmiştir. bir makine için (Abelson & Sussman, 1985; Shustek, 2006) .

Yazılım kaynak kodu, veri oluşturma ve toplama, veri görselleştirme, veri analizi, veri dönüştürme ve veri işlemeyi yalnızca bilimsel makalelerle karşılanmayan bir kesinlik düzeyinde uygular ve açıklar . Kamuya açık yazılımlar, bilimsel sonuçların çoğaltılmasını sağlar ve açık kaynaklı yazılım, yazılım kaynak kodunun korunmasını ve kalıcı bir şekilde referans gösterilmesini sağlamak için uygun mekanizmaların devreye sokulması koşuluyla , araştırmacıların başkaları tarafından elde edilen sonuçların üzerine inşa etmesine olanak tanır (Di Cosmo, Gruenpeter ve Zacchiroli, 2020a) .

Araştırmacılar giderek sonuçlarını özetleyen araştırma makaleleri ve bu sonuçları desteklemek veya göstermek için yazılımlar üretiyorlar. Bu ikinci aktivite , çalışmalarının önemli bir bölümünü temsil edebilir ve araştırmacılar meslektaşları ve kurumsal yetkililer tarafından değerlendirilirken uygun şekilde dikkate alınmalıdır (Alliez ve diğerleri, 2020a; Clément-Fontaine ve diğerleri, 2019) .

Son olarak, araştırmacılar tarafından geliştirilen yazılım kaynak kodu, yazılımı yürütmek ve bilimsel olarak anlamlı sonuçlar üretmek için gerekli olan, çoğu akademi dışında geliştirilen karmaşık yazılım bileşenleri ağının üzerinde yalnızca ince bir katmandır (K. Hinsen, 2019) : Örnek olarak, Şekil 1 , popüler `matplotlib` kitaplığını kullanmak için

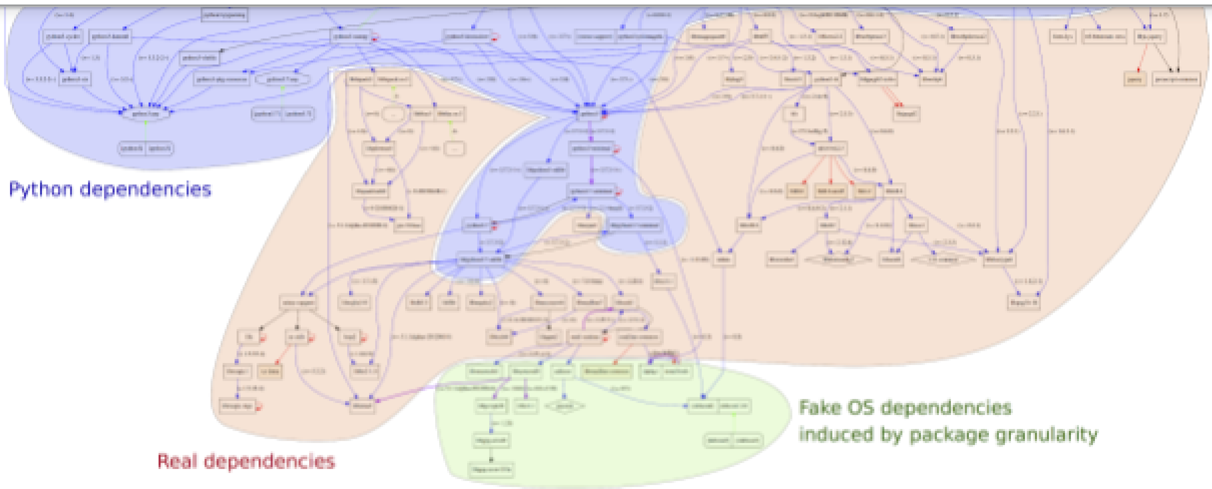


Figure 1. Example of the complexity in direct and indirect dependencies for a specific python package (matplotlib). Boxes represent actual packages (libraries that need to be installed on the system), arrows indicates dependencies to other packages, labels indicates the minimal/maximal version number. In blue the Python dependencies, in red the "true" system dependencies incurred by python (e.g., the libc or libjpeg62), in green some "fake" dependencies incurred by the package management system but which are very likely not used by python (e.g., adduser or dpkg).

Sonuç olarak, akademide yazılmış yazılım kaynak kodunu destekleyen bilimsel altyapılar, standartları benimsediklerinden ve dünya çapında on milyonlarca akademik olmayan yazılım geliştiricisi tarafından kullanılanlarla uyumlu mekanizmalar sağladıklarından emin olmak *için ekstra yol kat etmelidir* .

Yine de, araştırma faaliyetleriyle ilgili yazılımların kaynak kodunun uzun vadede korunmasını, düzgün bir şekilde tanımlanmasını ve tanımlanmasını ve buna katkıda bulunanlara verilen kredinin sağlanması için yeterli desteğin sağlanması söz konusu olduğunda yapılacak çok şey var (Toplantı, 2019) .

Onlarca yıldır , yazılım kaynak kodunun arşiv olması amaçlanmayan geliştirme platformları aracılığıyla kullanıma sunulduğunu ve bunlara bağlantılar kullanılarak araştırma makalelerinde referans verildiğini veya sadece adlarıyla anıldığını gördük (Howison & Bullard, 2015) . Hepimiz bu platformlara olan bağlantıların çürüyebileceğini (Spinellis, 2003) ve platformların kendilerinin ortadan kalkabileceğini biliyorduk ,-ancak son zamanlarda, Google Code gibi büyük platformların kapatılmasıyla tehlikenin boyutu anlaşılmaya başlandı. [1] ve Gitorious.org [2] ve Bitbucket [3] 'te Mercurial sürüm kontrol sistemi (VCS) desteğinin aşamalı olarak sona ermesi .

Sonuç olarak, milyonlarca yazılım projesi yerinden edildi veya kayboldu ve bilimsel bilgi ağı önemli ölçüde tehlikeye girdi.

İşte bu olgunun kendini nasıl gösterdiğini gösteren birçok örnekten birkaçı :

- MIT'den WorldView projesinin web sayfasındaki kaynak kodun bağlantısı artık Gitorious.org'daki uzun süredir devam eden bir depoya işaret ediyor: http://www.gitorious.org/worldview .
- Kaynak kodlarının güvenli olduğunu düşünen makalelerin yazarları, birkaç yıl sonra bu kodun kaybolduğunu keşfeder: örneğin, (Di Cosmo & Danelutto, 2020) veya bu son tweet'te nelerin rapor edildiğine bakın.



Araştırma dünyasında, genel olarak ama özellikle açık bilim için büyük önem taşıyan bu konuyu ele almak için bazı girişimler ortaya çıkmaya başladı. Zenodo (2014'te) gibi bilimsel depolar, araştırmacıların araştırma yazılımlarını manuel olarak veya doğrudan GitHub'dan otomatik olarak arşivlemeleri için bir mekanizma sağladı.

Software Heritage (Abramatic ve diğerleri, 2018a; Di Cosmo & Zacchiroli, 2017) , tüm yazılım kaynak kodlarını tam geliştirme geçmişiyle birlikte proaktif olarak toplama ve arşivlemenin ağır yükünü üstleniyor (neyse ki yukarıdaki tüm örnekler dahil). Farklı boyutlarda olabilen ve farklı kurumsal veya disiplin gereksinimlerine hitap edebilen tüm bilimsel havuzların Software Heritage ile düzgün bir şekilde arayüz oluşturması ve araştırmacılara bekledikleri ek işlevleri sunması ve araştırma makalelerinin yazılımın arşivlenmiş sürümüne atıfta bulunması önemlidir.

Nitekim, yukarıdaki örneklerdeki kaynak kodu artık korunmuş olsa da, araştırma makalelerinde orijinal geliştirme platformuna işaret eden bağlantılar artık kopmuştur. Kaynak kodun bulunma, yeniden kullanılma veya araştırma bulgularını desteklediğini gösterme olasılığı daha düşüktür ve araştırma makaleleri de içeriklerinin bir kısmını (kaynak kodu) kaybetmiştir. Bu nedenle, yazılım arşivlemesi kesinlikle gerekli olsa da, araştırma makalelerinde arşivlenmiş versiyona, araştırmanın kod tarafından desteklenmesini ve yeniden kullanım potansiyelinin en üst düzeye çıkarılmasını, özellikle de içsel tanımlayıcıların kullanımının genelleştirilmesini sağlayacak şekilde atıfta bulunulması da bir zorunluluktur. .

Araştırma içeriği içinde arşivlenmiş sürümleri arşivlemek ve bunlara referans vermek sadece bir başlangıçtır. Yazılımı tanımlamak için onlarca mevcut ontolojiyle başa çıkmak, gerekli tüm meta verilerin alıntı amacıyla kolayca erişilebilir olmasını sağlamak ve yazılım kredisi ve alıntı (referanslama araçları ve yayıncı evi stilleri).

Bu ihtiyaçların ele alınması *standartlar, araçlar, altyapılar, eğitim, erişim* ve *yayıncılık topluluğuyla katılım gerektirecektir* . Ayrıca hem *geliştirme, iletişim* ve *sosyal yardım çabaları* hem de *operasyonel maliyetler için uygun finansmana ihtiyacı var* .

2.1. Kapsam ve Hedefler

Bilimsel Araştırma Yazılımı Altyapıları (SIRS) TF, mevcut uygulamaları ve yaklaşımları araştırmak, en iyi uygulamaları ve açık sorunları belirlemek ve EOSC'nin *yazılım kaynak kodunu* bir yazılım kaynak kodu haline getirmesine izin verecek *küresel bir altyapı mimarisi için somut öneriler formüle etmek için açık bir misyonla bir araya getirildi.* makaleler ve verilerle eşit düzeyde bilimsel dünyada birinci sınıf vatandaş.

Açıkça belirtildiği gibi (Clément-Fontaine ve diğerleri, 2019) , " *yazılım, aynı derecede bir itici güç (bir araç olarak), bir sonuç (bir çözümün varlığının kanıtı olarak) olduğu için araştırma dünyasında hibrit bir nesnedir.) ve bir çalışma nesnesi (bir eser olarak). Bu özel durum, ortaya çıkardığı çeşitli konulara uyarlanmış stratejiler, araçlar ve prosedürler tanımlamamız gerektiği anlamına gelir. Bunlar, yazılım tasarımına ve üretimine yapılan katkılardan alıntıları, yazılımı içeren araştırma sonuçlarının tekrarlanabilirliğini ve oluşturulan yazılım mirasının daha geniş kullanımını ve uzun vadeli sürdürülebilirliğini içerir.* "

Yazılımın araştırma dünyasındaki bu çok biçimli doğası nedeniyle, " *araştırma yazılımı* " *terimi farklı araştırma topluluklarında çok farklı anlamlar taşıyabilir: bu raporda, bu terimi herhangi bir disiplindeki araştırmacıların hissedebileceği yazılımı* belirtmek için kullanacağız . bir araç, sonuç veya çalışma nesnesi olarak kabul *edilsin, bilimsel altyapı desteğine ihtiyaç duyar* [4] .

üstesinden gelinmesi gereken birçok zorlu zorluğun çok iyi farkında olduğumuzu netleştirmek önemlidir. sürdürülebilirliği için ele alınması gereken karmaşık organizasyonel, ekonomik ve strateji konularının yanı sıra.

Bu TF'nin çalışmasının odak noktası farklıdır, çünkü iki ana nedenden dolayı *araştırma dünyasında yalnızca yazılım kaynak kodunu kasten ele aldık*. Her şeyden önce, yazılımın kaynak kodu *insan tarafından okunabilir bilgidir* ve yürütülebilir dosyalardan çıkarılamayan ve ilgili yürütülebilir dosya artık çalıştırılamadığında bile anlaşılabilen değerli teknik ve bilimsel bilgileri içerir. İkincisi, yazılım kaynak kodunun ele alınmasının bilimsel altyapılar için ortaya çıkardığı sorunları doğru bir şekilde ele almak, bu raporda açıkça belirtildiği gibi başlı başına önemli bir zorluktur ve ilk önce buna odaklanarak eyleme geçirilebilir öneriler sunmak daha kolaydır.

2.1.1. Arşiv, Referans, Tanımlama, Kredi: Dört Sütun

Yukarıda gördüğümüz gibi, araştırma dünyasındaki *yazılım kaynak kodu* birkaç nedenden dolayı araştırma verilerinden oldukça farklıdır , bu nedenle popüler FAIR (Bulunabilirlik, Erişilebilirlik, Birlikte Çalışabilirlik ve Yeniden Kullanım) Kılavuz İlkelerinin araştırma verileri için (Wilkinson ve diğerleri, 2016) , bunun için tasarlanmadıkları için buna iyi uymuyor. Yazılımla uğraşırken kullanılmak üzere FAIR ilkelerinin nasıl değiştirileceğini, hatta tamamen elden geçirileceğini tartışmak amacımız değil: diğer çalışma grupları bu zorlukla boğuşuyor [5] .

Burada, yazılımları bilimsel dünyada ve bilimsel altyapıların önemli bir rol oynadığı birinci sınıf vatandaş yapmak için ele alınması gereken *dört temel somut soruna* odaklanıyoruz :

- Araştırma yazılımı eserlerinin kaybolmamasını sağlamak için şimdiye kadar geliştirilen **arşivleme yazılımları** (Abramatic ve diğerleri, 2018a) ;
- Araştırma yazılımı eserlerinin tam olarak tanımlanabilmesini sağlamak için **başvuru yazılımı** (Di Cosmo, Gruenpeter, & Zacchiroli, 2020a) ;
- Araştırma yazılımı eserlerini kolayca keşfetmek ve tanımlamak için yazılımı tanımlama **(** Bönisch ve diğerleri, 2012a; Di Cosmo ve diğerleri, 2019) ;
- Yazarlara uygun kredi verilmesini sağlamak için kredilendirme (**Alliez** ve diğerleri, 2020b) .

Bilimsel altyapılardan oluşan küresel bir mimarinin, aşağıda ARDC olarak da kısaltılan *dört sütun* olarak adlandırdığımız araştırma dünyasındaki *Arşivleme, Referans, Tanımlama ve Kredi yazılım kaynak koduna* uygun araçlar sağlaması gerektiğini *düşünüyoruz* .

Bu amaçla, tüm *arşivleri, yayıncıları* ve *toplayıcıları* (kataloglar dahil) kapsayan, *araştırmacılar tarafından yazılan yazılım kaynak koduyla ilgilenen, Avrupa genelindeki mevcut operasyonel altyapıların temsili bir panelini bir araya getirdik.*

Dört aylık ortak çalışma boyunca, bu temsilciler, bu sütunların inşasında yer alan bazı zorlukları ele alma konusundaki somut ve olgusal deneyimlerine dayanarak, söz konusu meseleler hakkında derinlemesine bir anlayış geliştirmeye yardımcı oldular.

2.2. TF'ye Katılan Altyapılar

SIRS TF, EOSC Architecture WG temsilcilerinden ve *araştırmacılar tarafından yazılan yazılım kaynak koduyla ilgilenen operasyonel altyapıların temsilcilerinden oluşur* . Aşağıdaki bölüm, katılımcı Avrupa altyapılarının tanıtımına ayrılmıştır. *Bu altyapılar, altyapının birincil amacına* bağlı olarak üç gruba ayrılır : *Arşivler* , *Yayıncılar* ve *Toplayıcılar* .

Bu rapor bağlamında, birincil hedeflerinden biri topladıkları dijital içeriğin *uzun vadeli korunması olan* hizmetleri belirtmek için *Arşivler terimini kullanıyoruz.* Bu, akademik dünyadaki *kurumsal arşivlerden* [6] *disiplin arşivlerine* [7] kadar geniş bir hizmet yelpazesinin yanı sıra daha geniş bir kapsamı olan hizmetleri içerir.

Yayıncılar , bir yayın üretmek ve yayma, tanıtım ve *arşiv sürecini* yönetmek için muhtemelen ilgili kaynak kodu ve verilerle birlikte gönderilen araştırma metinlerini hazırlayan kuruluşlardır . Yazılım ve veriler , derginin politikasına bağlı olarak ana yayının bir parçası veya *ek materyal olarak verilen varlıklar olabilir.* Ek olarak, yayıncılar, konuya özel bir uzmanlar topluluğu tarafından yürütülen, kabul edilen araştırma materyalinin (genellikle akran değerlendirmesi) kalitesini sağlamak için bir süreç uygular.

genellikle toplu içeriğin açıklamasına erişim sağlayan bir arama motoru sağlar ve başka bir yerde arşivlenmiş sürümlerine bağlantılar sağlayabilir. Bu hizmetler genel olabilir veya disipliner, coğrafi veya kurumsal bir kapsama sahip olabilir.

Aşağıdaki özet sayfaları , SIRS TF'de temsil edilen dokuz altyapıyı sunar: üçü *arşiv* kategorisi (HAL, Software Heritage ve Zenodo), üçü *yayıncı* kategorisi (Dagstuhl, eLife ve IPOL) ve *üçü* toplayıcılar için kategori (OpenAIRE, ScanR ve swMath).

2.2.1. Arşivler

Aşağıdaki liste, SIRS TF'de temsil edilen arşiv setini sağlar:

- HAL
- Yazılım Mirası
- Zenodo

Hizmetlerin coğrafi kapsamları (ulusal ve uluslararası) açısından büyük farklılıklar gösterdiğinden, bu altyapılar, Arşiv alt grubu ortamının saygın bir temsilidir ; içerik (evrensel ve disiplinsel) , boyut, kayıtlı kullanıcı sayısı, ele alınan yazılım projesi sayısı bakımından kapsam; ve iş akışı.



HAL

Archive ouverte

[illegible]

2001

[illegible]

610

[illegible]

FTE:20

[illegible]

Organization Overview

[illegible]

Scope

[illegible]

Workflow

[illegible]



Software Heritage

Nonprofit Foundation

2015

140 Million

Current: 14 FTE, 1,000,000€/year

Target: 50 FTE, 10,000,000€/year

Organization Overview

Scope

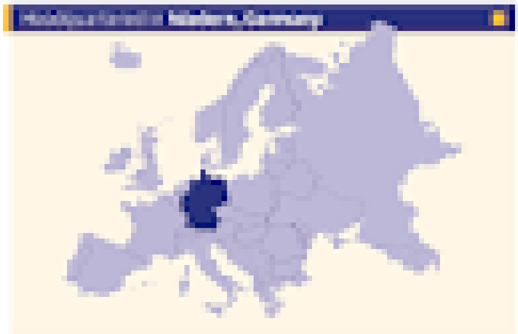
Workflow



2.2.2. yayıncılar

Aşağıdaki liste, SIRS TF'de temsil edilen yayıncı grubunu sağlar:

- Dağstuhl
- e-hayat
- IPOL



Dagstuhl

Open access editorial infrastructure

1990

100

FTE: 41

(Publishing 3 FTE)

Annual budget:

~3 Million EUR

Complete

Dagstuhl incl.

Seminar Center

Organization Overview

Scope

Workflow



eLife

Nonprofit organization

2012

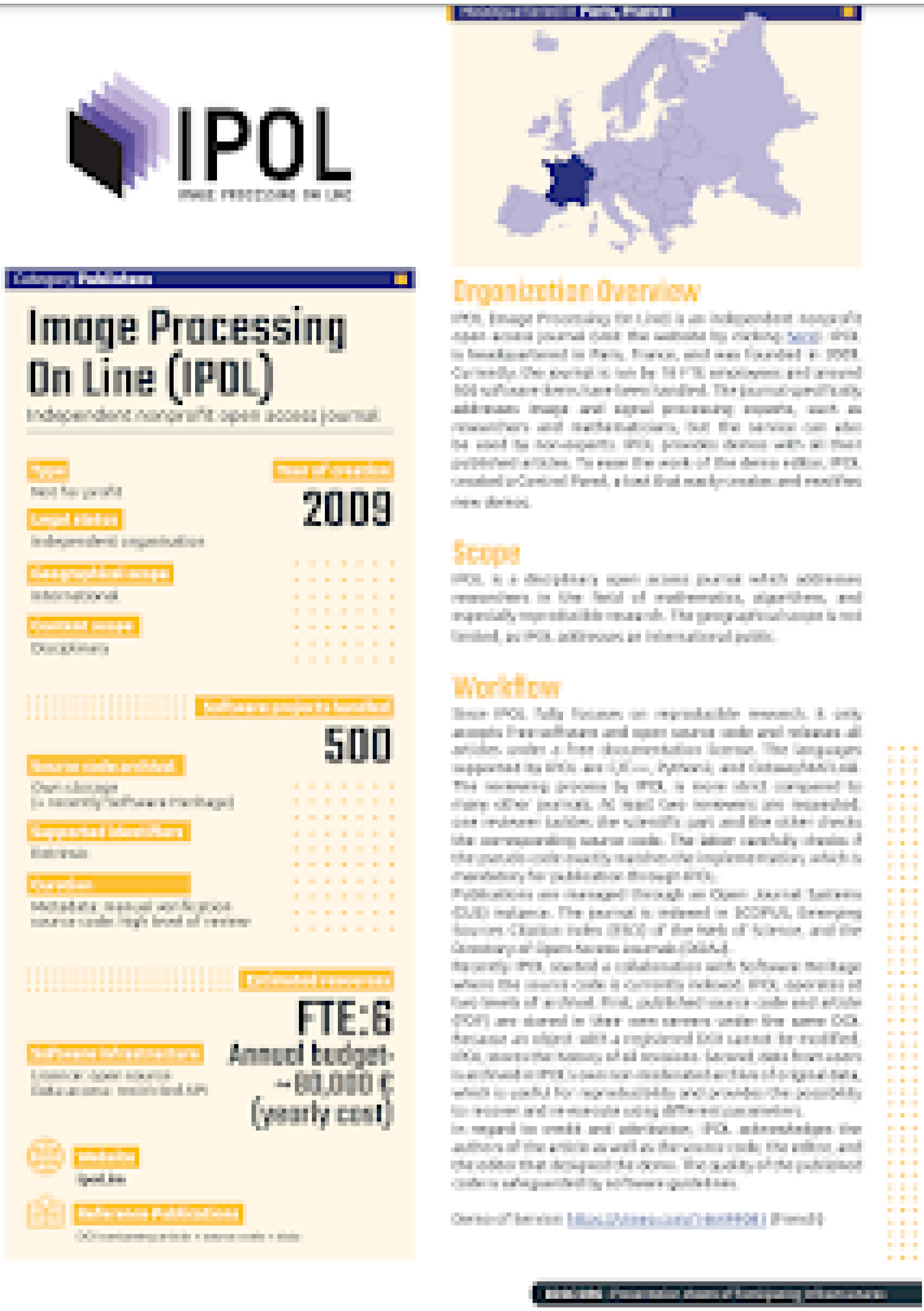
819

FTE:50

Organization Overview

Scope

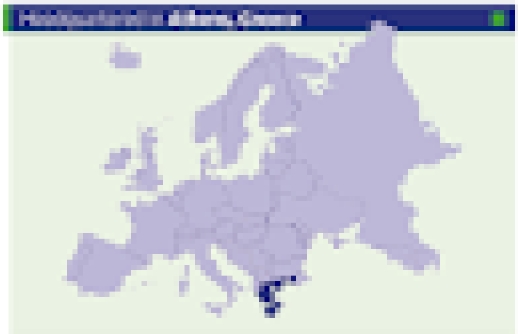
Workflow



2.2.3. toplayıcılar

Aşağıdaki liste, SIRS Görev Gücü'nde temsil edilen kataloglar da dahil olmak üzere toplayıcılar kümesini sağlar.

- OpenAIRE
- tarama
- swMATH.org



OpenAIRE

Independent non profit company

2019

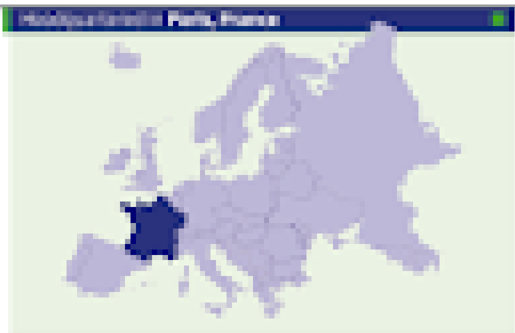
200,000

FTE:25
per year
(rough estimate)

Organization Overview

Scope

Workflow



scanR

Search engine

2016

400

FTE:1.5

Annual budget

€ 100.000

Organization Overview

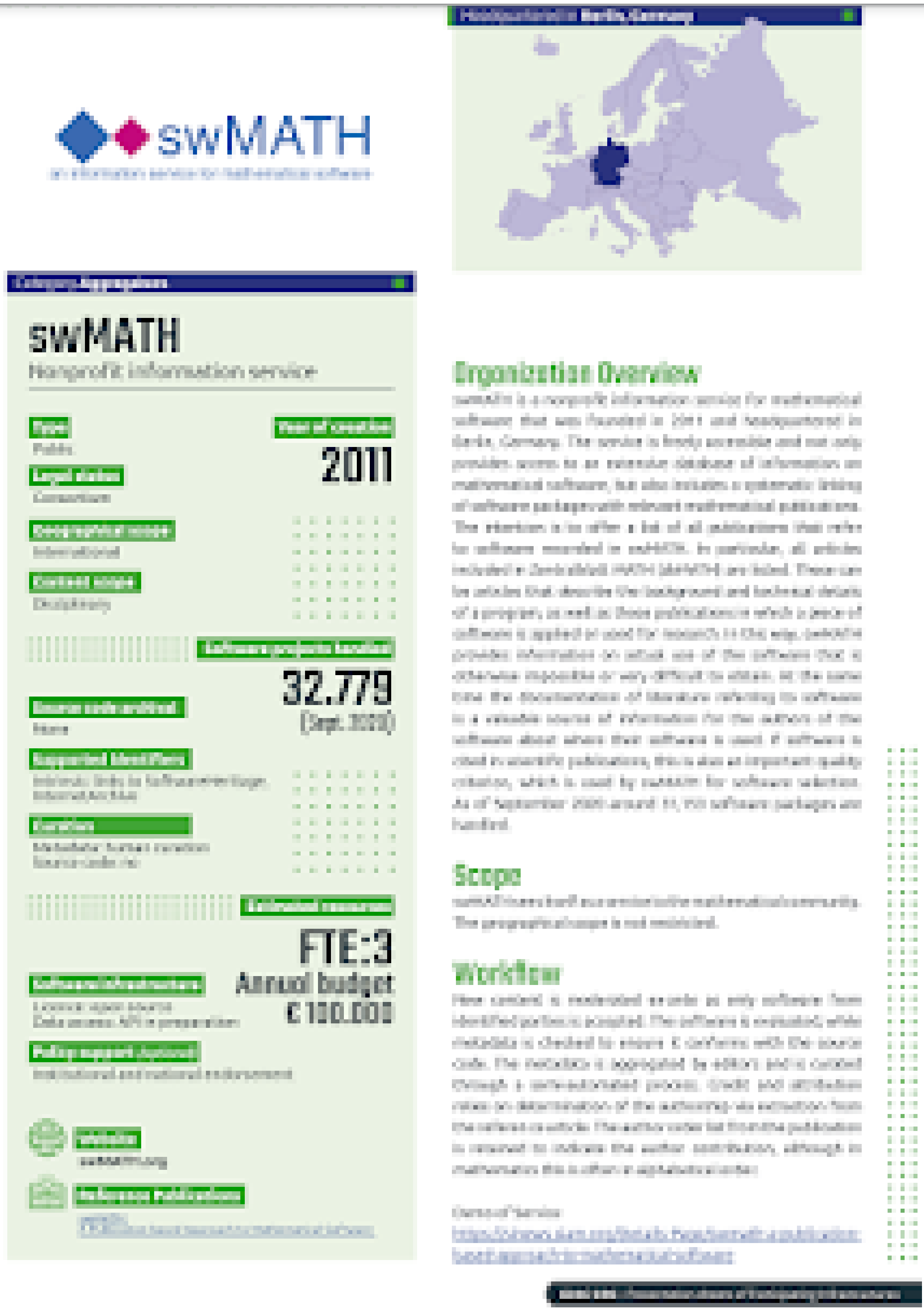
[illegible]

Scope

[illegible]

Workflow

[illegible]



3. Son Teknoloji

3.1. İlgili Girişimler ve İlgili Çalışmalar Hakkında Anket

Yazılım, on yıllardır araştırmalarda önemli bir rol oynamıştır ve Inria gibi bazı araştırma kurumlarında (Alliez ve diğerleri, 2020a) veya Hesaplamalı Fizik (Roberts, 1969) , Sayısal Hesaplama (Press, 1986) , Neurosciences (McDougal ve diğerleri, 2016) ve Astrophysics (Allen & Schmidt, 2015) . Serbest Kaynak'ın gelgit dalgası, endüstriyi Açık Kaynak yazılımı olarak kasıp kavurmadan önce, akademide de doğdu. *Ancak , bir araştırma çıktısı olarak* yazılımın önemine

Kapsamlılık iddiasında bulunmadan, bu uyanışın birkaç dikkate değer erken sinyali, örneğin Science Code Manifesto'da (Barnes, 2010) , 2010'da Birleşik Krallık'ta Yazılım Sürdürülebilirlik Enstitüsü'nün oluşturulmasında, 2010'da IPOL dergisinin oluşturulmasında bulunabilir. 2009 (Colom ve diğerleri, 2015) ve 2011 yılında ESEC/FSE Konferansı [8] için Andreas Zeller tarafından Carlo Ghezzi ve Shriram Krishnamurthi ile tanıtılan ve şu anda Bilgisayar Bilimleri konferanslarında yaygın olan ilk Artifakt Değerlendirmesi (Childers ve ark. , 2016; Krishnamurthi, 2011) ve ACM Rozetleme şemasının onaylanmasına yol açtı (ACM, 2016) .

2015 civarında, birçok farklı alan ve disiplinde tekrarlanabilirlik için yazılımları korumanın ve bunlara atıfta bulunmanın önemini vurgulayan çok sayıda makale vardı (Allen ve diğerleri, 2017; Baker, 2016a; Collberg & Proebsting, 2016; Gil ve diğerleri, 2016; Krishnamurthi & Vitek, 2015) ve kaynak kodunu arşivlemek ve kalıcı tanımlayıcıları ve bilimsel meta verileri kaydetmek için GitHub/Zenodo entegrasyonu gibi bir değişiklik yapmaya başlamak için girişimler başlatıldı (*Making Your Code Citable · GitHub Guides* , s. d.) , CodeMeta girişimi (Jones ve diğerleri, 2016) , yazılım atıfları üzerine FORCE11 çalışma grubunun oluşturulması (Smith ve diğerleri, 2016) , DARTS eser serisi (Wagner, 2017)ve yazılım kaynak koduyla ilgili BKA ilgi grubu (Research Data Alliance/FORCE11 Software Source Code Identification WG ve diğerleri, 2020) .

Yazılıma olan ilgi yeni yeni artmaya başlarken, araştırma topluluğu *araştırma verileri* üzerinde tam hızla ilerlemiş ve araştırma verileri için *FAIR ilkelerinin* (Wilkinson ve diğerleri, 2016) Eylül ayı boyunca en yüksek düzeyde onaylandığı noktaya gelmiştir. 2016 G20 toplantısı [9] . Olayların bu kronolojisi ve yazılımın hâlâ büyük ölçüde bir araç veya yalnızca başka bir veri parçası olarak görülmesi gerçeği, yazılımın , muhtemelen bazı küçük değişikliklerle, FUAR ilkelerine nasıl uyabileceğini görmek için neden önemli miktarda enerji harcandığını açıklayabilir . (Katz & Clark, 2019; Lamprecht et al., 2020) , yazılıma yeniden uyarlanmış ilkeler geliştirmek yerine.

Yazılım geliştirme, araştırmada önemli bir rol oynar, bu nedenle , Fransa'daki DEVLOG ağı gibi, bazı ülkelerde oldukça uzun bir süredir araştırma topluluğu içinde çalışan yazılım geliştiricilerini birleştirme ve destekleme çabaları olması şaşırtıcı değildir [10] . Daha yakın zamanlarda, Araştırma Yazılım Mühendisi (RSE) terimi, Avrupa'da ve ötesinde, araştırma yazılımı geliştirme becerilerine sahip bireyleri bir araya getiren, RSE'lerin tanınmasını savunan ve araştırma yazılımının bir ilk olmasını teşvik eden çeşitli ulusal ve çok uluslu girişimler tarafından benimsenmiştir. akademide sınıf vatandaş (de-RSE, NL-RSE, Nordic-RSE, UK-RSE) [11] .

Son yıllarda, genel olarak yazılımın ve özel olarak yazılım kaynak kodunun nihayet, Fransız Ulusal Açık Bilim planından (Clément-Fontaine ve diğerleri, 2019) yazılım kaynağına ilişkin Paris Çağrısı'na kadar politika belgelerinde bahsedilmeye başladığını gördük . Inria-UNESCO uzman grubu toplantısı tarafından yayınlanan kod (E. grup toplantısı UNESCO, 2019) .

Bu raporda, araştırmada kullanılan yazılım kaynak kodu için bilimsel mimarideki üç temel bileşene özellikle odaklanıyoruz : *arşivler, yayıncılar ve toplayıcılar* . Aşağıda, bu alanlardaki ilgili girişimlere ve çalışmalara kısa bir genel bakış sunuyoruz.

3.1.1. Arşivler

Bilgisayar Tarihi Müzesi ve benzeri kuruluşlar gibi, özellikle bilgi işlem tarihine adanmış girişimler tarafından öncülük edilen *yazılım programlarını koruma faaliyeti* , arşivler tarihinde nispeten yeni bir endişedir ve esas olarak, *fiziksel medyanın* arşivlenmesine odaklanmıştır. bu yürütülebilir programlar dağıtıldı (disketler, oyun kartuşları, CD'ler veya DVD'ler), bunlar kataloglandı ve tıpkı kitaplar gibi saklandı. Bu alandaki dikkate değer bir aktör Fransız milli kütüphanesidir (BNF), Fransa'da olduğu gibi, diğer ülkelerden farklı olarak, yazılım programları 1992'den beri yasal mevduata tabidir [12] . Daha yakın zamanlarda, sanallaştırma ve öykünme teknolojilerinin ortaya çıkması ve metalaşmasıyla birlikte odak,özellikle dijital varlıklara erişimi korumanın bir yolu olarak *eski yazılım programlarını daha yeni makinelerde çalışır durumda tutmak* [13] .

Yazılım kaynak kodunu kendi başına korumaya olan ilgi , 2006 yılında Len Shustek'in kristal berraklığında ufuk açıcı bir makalesinde çok önemli bir konu olarak tanımlanmasına rağmen çok daha yenidir (Shustek, 2006) . Yazılım kaynak kodu, yalnızca birkaç yıl öncesine kadar, mevcut bilimsel arşivler ve havuzlar, benzer bir dışsal kalıcı tanımlayıcı atanan kaynak kodu paketlerinin depolanmasına izin vermeye başlayıncaya kadar, büyük ölçüde bilimsel havuzların ve kurumsal ve ulusal arşivlerin kapsamı dışında kabul edildi. veri kümeleri için kullanılanlara: örnek olarak, DataCite'ta bir yazılım paketi için kaydedilen ilk DOI (Paskin, 2010) yalnızca 7 Eylül 2011'e aittir (Fenner ve diğerleri, 2018) ve*yazılım kategorisi Zenodo'da* yalnızca 2014'te ve HAL'de yalnızca 2018'de tanıtıldı (Barborini ve diğerleri, 2018; Di Cosmo, Gruenpeter, Marmol, ve diğerleri, 2020) . Bu bilimsel veri havuzları, depolanan yazılım paketlerine erişim kontrolü, meta veri güncelleme mekanizması, eş-inceleme anonim erişim ve isteğe bağlı olarak depozitoyu denetleme veya

otomasyonundan sonra özellikle kod paketlerini yatırmak için en popüler hizmettir ; ancak, kaynak kod demetleri biçimindeki yazılım mevduatları, kendi verilerini kontrol etmek isteyen farklı bilimsel havuzlarda veya dijital arşivlerde çeşitli biçimlerde kullanıma sunulmuştur.

Bu arada, yazılım geliştirme son yarım yüzyılda katlanarak büyüyor ve onu destekleyen araçlar ve platformlar, *orijinal içeriğin 22 ayda bir ikiye katlanması* ve *orijinal taahhütlerin her 30 ayda bir ikiye katlanmasıyla hızlı bir* şekilde gelişiyor . Şekil 2, farkında olduğumuz yazılım geliştirme evriminin en geniş analizinden alınmıştır (Rousseau ve diğerleri, 2020) .

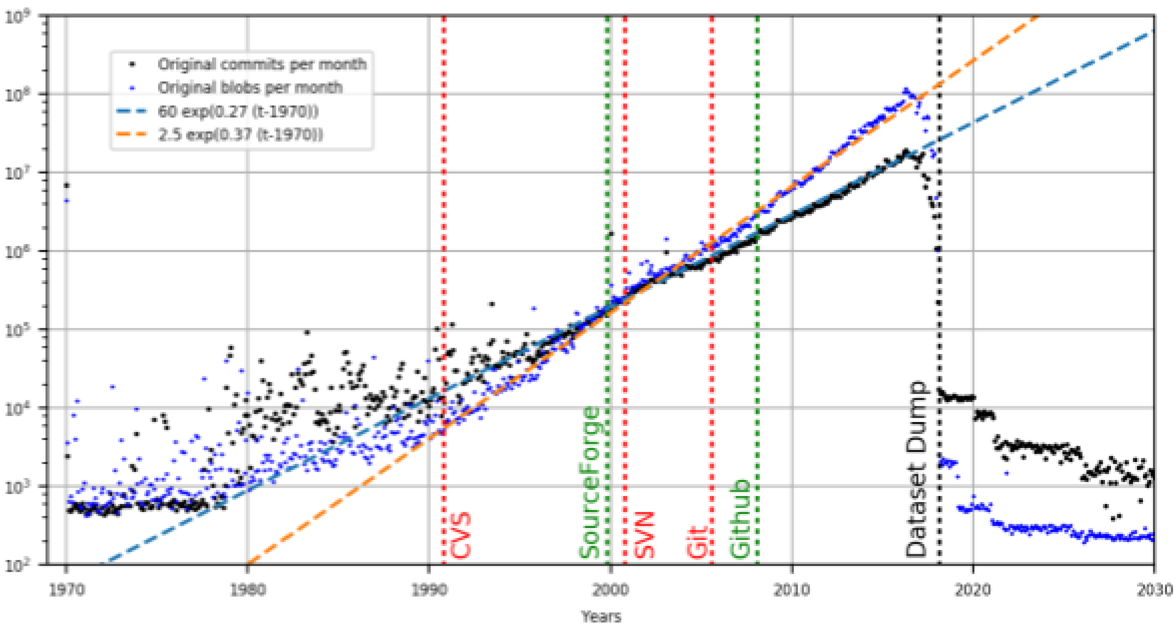


Figure 2. Global production of original software artifacts over time, in terms of never-seen-before revisions and file contents (lin-log scale). Major events in the history of version control systems and development forges are materialised by vertical bars.

Onlarca yıldır yazılım kaynak kodu, dünya çapındaki yazılım geliştiricileri tarafından esas olarak iki kategoriden birine giren platformlar aracılığıyla sağlandı:

İşbirlikçi geliştirme platformları

Yazılım projelerinin oluşturulmasına ve ortak geliştirilmesine olanak sağlayan hizmetler; birkaç iyi bilinen örnek SourceForge, Gitlab'dır. com , GitHub ve BitBucket.

D dağıtım platformları

Esas olarak yazılım paketlerinin (sürümlerinin) dağıtılması için kullanılan hizmetler; iyi bilinen birkaç örnek CPAN, CRAN, CTAN, PyPI ve NPM'dir.

Google Code ve Gitorious.org gibi büyük geliştirme platformlarının 2015'te aniden kapanması [14] , *bir buçuk milyondan fazla yazılım projesini* tehlikeye atarak, hepimize acımasızca hatırlattı ki, şaşırtıcı görünse de, bu platformlardan hiçbiri birincil olarak tasarlanmamıştır. bir *arşiv olarak* , yazılım kaynak kodunu geliştirme geçmişiyle birlikte uzun süre korumak anlamına gelir. *Bu olaylar, kamuya açık tüm yazılımların kaynak kodunu, tüm geliştirme geçmişiyle birlikte* toplama, koruma ve paylaşma misyonuyla *evrensel bir yazılım kaynak kodu arşivi* geliştirmeye acil ihtiyacı ortaya çıkardı .. Koşulların bir kombinasyonunun sonucu olarak, tam olarak bu misyona sahip altyapılar, Inria tarafından UNESCO ile işbirliği içinde başlatılan açık, kar amacı gütmeyen bir girişim olan Software Heritage tarafından birkaç ay önce inşa edilmeye başlandı (Abramatic ve diğerleri, 2018a) . Tüm bu projeler zamanında kurtarıldı.

Software Heritage, Git veya Mercurial gibi modern dağıtılmış VCS'lerde bulunan aynı teknolojiyi kullanarak, *tüm geliştirme geçmişiyle birlikte tüm yazılım kaynak kodunu* uzun vadede korumak için özel olarak tasarlanmış tek altyapıdır (Di Cosmo ve diğerleri, 2018; Di Cosmo, Gruenpeter , & Zacchiroli, 2020b; Di Cosmo & Zacchiroli, 2017) ve bilimsel yayınlarda kaynak kodunu arşivlemek ve referans göstermek için basit ve genel mekanizmalar sağlar (Di Cosmo, 2020b) .

3.1.2. yayıncılar

Buckheit, Donoho ve diğerleri , bilimsel araştırmalardaki *güvenilirlik krizi hakkında* uyardıklarından (Buckheit & Donoho, 1995) , tekrarlanabilirlik sorunu farklı alanlardan binlerce bilim insanı tarafından doğrulanmıştır (Baker, 2016b) . Bu sorun birçok araştırma disiplininde ortaktır, ancak özellikle çoğu durumda araştırmanın sonucunun bir algoritma olduğu ve bu nedenle tekrarlanabilirliği ele almak için hiçbir mazeret olmadığı hesaplamalı bilimlerle ilgilidir .

hesaplamayı içeren her türlü araştırmayı ilgilendirdiğini unutmayın. Örneğin, veri toplamak ve sonuçları sunmak için rakamlar oluşturmak için kendi yazılımlarını yazabilen sosyal bilimler üzerine yayınlar veya beşeri bilimler dergileri.

Bu krize farklı düzeylerde çözümler getirmeye çalışmak için çeşitli girişimler geliştirilmiştir. Yazılım eserlerinin meslektaş incelemesi, eser değerlendirme komiteleri ve rozet şemaları aracılığıyla tanıtıldı (ACM, 2016; Childers ve diğerleri, 2016; Krishnamurthi, 2011) ve yayıncılar tarafından bu eserleri arşivlemek ve referans vermek için çeşitli çözümler önerildi. , ya araştırma makalesiyle ilişkili yardımcı materyal olarak depolanmaları yoluyla (ACM, s. d.) ya da yazılımın yayınlanması veya kendi başlarına değerlendirilen araştırma eserlerinin (Seinstra ve diğerleri, 2015; Smith ve diğerleri, 2018; Wagner, 2017)) . IPOL gibi öncü dergiler (Arévalo ve diğerleri, 2017; Colom ve diğerleri, 2015) çok daha ileri gitti, belirli görüntü işleme alanında algoritmaları çalıştırmak, değerlendirmek ve karşılaştırmak için altyapılar kurdu.

Araştırma makaleleriyle ilişkili *yürütülebilir dosyaların tam olarak yeniden üretilebilirliğinin* kutsal kâsesi, zaman içinde geliştirilen, sanallaştırma veya kapsayıcı ortamlarından dizüstü bilgisayarlara ve tam olarak yürütülebilir olanlara kadar çok çeşitli araçlarla birçok disiplinde oldukça fazla ilgi odağı olmuştur. araştırma makaleleri. Mevcut çözümlere ve bunların avantajlarına ve sınırlamalarına ilişkin eleştirel bir inceleme bu raporun kapsamı dışındadır, ancak burada bu çabaların, ilgili kaynak kodunun tamamını uygun şekilde arşivleme ve referans alma ihtiyacını *tamamlayıcı* olduğunu ve onun yerine geçmediğini vurguluyoruz.

Kaynak kodu ve veriler yayının temel bir parçası olduğunda, araştırma çalışmasının kendisinin bir parçası olduğu zaman, çeşitli girişimler *artırılmış yayınlara* doğru ilerler (Barnes, 2010; Clément-Fontaine ve diğerleri, 2019; Colom ve diğerleri, 2019). metin, kaynak kodu ve verileri bir bütün olarak birleştirmek (Borgman ve diğerleri, 2012) ve yazılım kaynak kodunun yazarlarına hak ettikleri itibarı vermek (Alliez ve diğerleri, 2020a; CASRAI, 2015; Di Cosmo, Gruenpeter, & Zacchiroli, 2020b; Smith ve diğerleri, 2016) . Bu , bu maddelerin araştırma çalışmasının önemli bir parçası olarak kabul edildiği göz önüne alındığında (Allen ve diğerleri, 2017; Childers ve diğerleri, 2016) , araştırmanın değerlendirilme biçimindeki değişiklikleri ifade eder (E. grup toplantısı UNESCO, 2019). Bu yeni tür yayıncılar, açık veri havuzları (Bönisch ve diğerleri, 2012b) ve kalıcı arşivler (Abramatic ve diğerleri, 2018b; Di Cosmo, Gruenpeter, Marmol ve diğerleri, 2020) ile sıkı bir şekilde çalışmaya ve eser alışverişi yapmaya teşvik edilmektedir. standartlaştırılmış şemalarla otomatik olarak meta veriler (Burton ve diğerleri, 2017) .

3.1.3. toplayıcılar

Araştırma yazılımı için toplayıcılar , kalıcı tanımlayıcılar, indirme URL'leri, özetler, katkıda bulunanlar, diğer araştırma ürünlerine bağlantılar vb. gibi yazılım hakkında ve bazı durumlarda kaynak koduyla ilgili üst veri bilgilerini toplar. kılavuzlara, havuzlara ve hatta diğer toplayıcılara referans makaleleri.

Araştırma yazılımı odaklı toplayıcılar 30.000'den fazla küratörlü girişe sahip Matematik için swMath hizmeti (Bönisch ve diğerleri, 2012b) veya 2.000'den fazla küratörlü girişe sahip Astrofizik için ASCL hizmeti (Allen & Schmidt, 2015) gibi belirli disiplinlerdeki araştırma yazılımı için toplayıcılar) en uzun süredir var, ancak yeni ve canlı girişimler, 34.000'den fazla girişle Makine Öğreniminde kod girişimi olan Kağıtlar gibi daha yakın zamanda hayata geçti (Stojnic ve diğerleri, 2019) .

Toplanan bilgilerin kalitesini sağlamak için, ASCL for Astrophysics (Allen & Schmidt, 2015) gibi diğer girişimler bir tür yayın kuruluna güvenmektedir . Bir başka dikkat çekici örnek ise araştırma faaliyetleri için yararlı olan yazılımlar hakkında bilgi toplamak amacıyla Plume projesi tarafından oluşturulan katalogdur (Plume, 2013) : başarıyla dağıtılan ve kullanımda olan yazılım projeleri hakkında manuel olarak küratörlüğünü yapılan 400'den fazla girişten oluşan bir koleksiyon tutar. en az üç farklı araştırma laboratuvarında

Araştırma yazılımı bağlam içi toplayıcılar Yukarıda bahsedilenler gibi yazılıma özel toplayıcılar, özellikle bir yazılım girdileri koleksiyonunun küratörlüğüne odaklanır, ancak yazılımlar, ilk etapta yazılımı ele almak için özel olarak tasarlanmamış veya oluşturulmamış toplayıcıların da dikkatini çekmektedir.

Bu kategorideki dikkate değer bir aktör, 2011'den beri metadata şemasına yazılım dahil eden ve Zenodo gibi mevduatları için DOI'ler sağlayan bilimsel havuzlarla çalışan, bilimsel içerik için bir DOI kayıt ajansı olan DataCite'dir . FORCE11 yazılım atıf ilkelerinin yayınlanmasından sonra (Smith ve diğerleri, 2016) , şemayı 2017 yılında daha da güncellemiş, özellikle araştırma yazılımlarına odaklanmıştır (DataCite Metadata Çalışma Grubu, 2017) ve Ekim 2020 itibariyle yaklaşık 150.000 adettir. Çoğunlukla Zenodo/GitHub entegrasyonu aracılığıyla, yaklaşık 40.000 farklı yazılım projesinin farklı yazılım sürümleri için kayıtlı DOI'ler (Fenner ve diğerleri, 2018). Bu çabanın ana odak noktası, yazılım ürünlerine yapılan tüm referansları tek bir kayıt defterinde, DataCite Meta Veri Deposu'nda toplamak, bunları DataCite Commons'a dahil etmek (Fenner, 2020) veya bunları, aşağıdakiler gibi diğer genel toplayıcıların kullanımına sunmaktır. OpenAIRE, DataCite meta veri formatında veya CodeMeta veya schema.org dahil diğer formatlarda (Burton ve diğerleri, 2017) küratörlüğü ve kalite kontrolünü diğer aktörlere bırakır.

genişletmeye ilgi duyduğunu ifade ederken (Peroni & Shotton, 2020) , OpenAIRE zaten yazılımla ilgili meta verileri diğer bilimsel kaynaklara ve aktörlere OpenAIRE Araştırma Grafiğine bağlamak için topluyor (K. Hinsen, 2019) Bugün, OpenAIRE hasat edilen 198.000 yazılım varlığını sayıyor Biotools, Zenodo, Figshare, DOE CODE, EGI Uygulama Veritabanı, DataCite kaynaklarından, bunlar sırasıyla SoftwareHeritage kalıcı tanımlayıcılarına ve orijinal yazılım havuzlarına (örn. GitHub, BitBucket, vb.) bağlıdır. İlginç bir şekilde, OpenAIRE yayınlar ve yazılımlar arasında (hasat ve tam metin madenciliği yoluyla) anlamsal bağlantılar ve araştırma projeleri ile yazılımlar arasında yaklaşık 1.850 bağlantı belirlemiştir.

Toplayıcıların katma değeri Toplayıcıların katma değeri iki yönlüdür. Bir yanda, aksi takdirde bağımsız olarak erişilmesi gereken farklı kaynaklardan gelen bilgileri bütünleştirme, uyumlu hale getirme ve bunlara erişim sunma yeteneği. Öte yandan, birleştirilmiş içeriği orijinal kaynaklarda bulunmayan bilgilerle zenginleştirme yeteneği. Örneğin, yazılımı ilgili araştırma literatürüyle ilişkilendirmek için çeşitli yaklaşımlar kullanılmıştır: bazı iş akışları tamamen manuel olarak gönderme ve iyileştirmeye dayanırken, diğerleri aynı zamanda akademik literatürden yazılıma referanslar çıkarmaya yardımcı olmak için çeşitli derecelerde gelişmiş araçlar kullanır. Disiplin. Bilimsel bir çalışmada yazılımdan bahsetmenin alabileceği çeşitli biçimler nedeniyle, tam otomatik referans çıkarma, sağlam ve eksiksiz olmaktan uzaktır.(Howison & Bullard, 2015) ve devam eden farklı deneyler var. Örneğin, Asclepias projesi yazılıma referanslar çıkarmak için PubMed, NASA ADS ve CrossRef/DataCite gibi alana özgü ve genel toplayıcılara güvenirken (Muench ve diğerleri, 2017) , kod içeren makaleler doğal dil işlemeyi ve makine öğrenimini kullanmayı önerir. bir topluluğun bu yazışmayı oluşturma çabasına yardımcı olmak için (Stojnic ve diğerleri, 2019) , OpenAIRE bilinen yazılım havuzlarına (örn. BitBucket, GitHub, vb.) URL'leri tanımlamak için tam metin madenciliğinden yararlanır ve bunları makale PDF'lerine ilişki olarak ekler. SoftwareHeritage.org'a ilişkin kalıcı URL ile birlikte makale meta verileri.

Bu yaklaşımlar, ilgili araştırma yazılımını belirlemek için araştırma yayınlarını bir hakikat kaynağı olarak kullanır ve özellikle swMath örneğinde olduğu gibi araştırma yayınlarının kalite iyileştirme sürecinden yararlanırken kaliteli değerli bilgiler üretebilir (Bönisch ve diğerleri, 2012b) . Yayıncı, doğrudan makalenin meta verilerine, kullanılan, üretilen veya yayında bahsedilen yazılım projeleri için yazılım projesinin bahsedildiği ayrıntı düzeyine ilişkin uygun açıklama ile kalıcı bir tanımlayıcı dahil ederse, iş akışları basitleştirilebilir. Bu bilgiler, yayıncı tarafından Crossref'e gönderilen meta verilere veya yayıncı tarafından PubMed Central gibi tam metin arşivlerine yerleştirilen JATS meta verilerine dahil edilebilir.

Son olarak, toplayıcıların yazılıma artan ilgisi, aynı zamanda Açık Bilim uygulamasını geliştirmek için büyük umutları ve metriklerin yazılım üretimi veya alıntı üzerinde kaçınılmaz olarak ortaya çıkmasının potansiyel olarak yıkıcı etkileri nedeniyle büyük zorlukları beraberinde getiriyor. bilim dünyasında olabilir. Bir yandan, örneğin Open Science Monitor'de önerilen bazı metriklerin kalitesi büyük endişeleri gündeme getirdi (Dacos ve diğerleri, 2018) . Öte yandan, yazılım için tamamen sayısal göstergeleri teşvik etmeden önce, özellikle de DORA gibi nitel kriterlere nicel kriterlere değer veren ilkeler etrafında büyüyen bir uluslararası fikir birliği ortaya çıktığı için, tüm sonuçları dikkatlice düşünmek gerekir (*San Francisco Araştırma Değerlendirme Deklarasyonu (DORA)* , 2013) veya Hong Kong ilkeleri (Moher ve diğerleri, 2020) .

3.2. Gruptaki Son Teknoloji Sunumlarının Özeti

Bu bölümde, her bir katılımcı altyapının uygulamalarının sunumlarının ateşlediği değişimden ortaya çıkan temel bulguları özetliyoruz.

3.2.1. Arşivler

Araştırma yazılımlarının arşivlenmesi, araştırma ürünlerinin kaybolmamasını sağlamak için önemlidir. Şu anda, açık bilim için meta veri ve PID standartları gibi belirli keşiflere ve farklı arşivleme stratejilerine sahip çeşitli arşivler mevcuttur ve kısmen birbiriyle bağlantılıdır.

Software Heritage, yalnızca araştırma yazılımını değil, genel olarak kaynak kodunu da bir araya getirir. Sistematik hasattan yararlanır, bu nedenle açık bir depozito gerekmez ve tüm VCS'leri kaynak ve izlenebilirlik ile tek tip olarak temsil eder. Yazılım için dahili tanımlayıcılar, proje durumu, proje sürümü, kaynak kodu durumu ve kod parçası gibi tüm ayrıntı düzeylerini kapsayan, arşivdeki 20 milyardan fazla yazılım eseri için sistematik olarak kullanılır.

HAL ise araştırma yazılımına özgüdür ve depozito gerektirir. Hizmet, kapsamlı, yazılıma özgü meta verilere sahiptir ve meta verilerin dikkatli ve manuel olarak iyileştirilmesi için insan müdahalesini içerir. Dışsal tanımlayıcılar meta verilere atanır, ancak içsel tanımlayıcılar, Software Heritage'a aktarım yoluyla yazılımın kaynak kodunun kendisine atanır.

Zenodo, esas olarak araştırma yazılımlarını bir araya getirir. HAL gibi depozito gereklidir, ancak GitHub kullanıcıları için Zenodo ile senkronizasyonu açıkça sağlayan para yatırma otomasyonu mümkündür. Meta veriler sahibi tarafından

Mevcut yazılım arşivlerindeki çeşitlilik göz önüne alındığında, yeni altyapılara acil bir ihtiyaç olmadığını, ancak yukarıda belirtilen arşiv ve havuzların ve var olan veya oluşacak daha birçoklarının birbirine bağlanmasına öncelik verilmesi gerektiğini düşünüyoruz. Bir sonraki adım, örneğin meta veri çıkarmayı otomatikleştirmek, yazılım meta veri standartlarını uyumlu hale getirmek, meta verilerin insan tarafından küratörlüğünü desteklemek ve metrikleri desteklemek için mevcut altyapıların işlevlerini genişletmektir. Araştırmacılar için yönergeler, önemli bir konu olarak depolama ve arşivden açıkça bahsetmelidir.

3.2.2. yayıncılar

Geçtiğimiz birkaç yıl içinde birkaç yayıncı, bilimsel çıktılar için baskın yayın modeli olarak açık erişime geçiş çabalarına öncülük etti. Bu aynı zamanda yazılımın yayılması için baştan itibaren adil ve uygun fiyatlı koşullar için bir yol açar, ancak uzman dergiler dışındaki yazılımlara verilen destek hala sınırlıdır. Katılımcı altyapılar, günümüzün yayıncılık işinde görüldüğü gibi, çok çeşitli kapsamları ve stratejileri yansıtmaktadır.

Dagstuhl, genel olarak yazılım için açık destek yerine, metinsel katkıyı tamamlayan eserlere odaklanır. Yazılım kaynak kodu, Dagstuhl'un kendi deposu kullanılarak arşivlenir. Yazılım yapıları, kendi meta verileri ve DOI'leri ile ilgili makaleden ayrı olarak yayınlanır. Tüm yazılım yapıları bir yapı değerlendirmesinden geçer ve meta veriler manuel olarak düzenlenir.

eLife dergisi, yazılım yazarlarına kredi vermenin yanı sıra yazılımın kullanımına ve yeniden kullanımına odaklanan açık bilim ve yeniden üretilebilirlik standartlarını uygulamıştır. Bir makale için oluşturulan kaynak kodunun izinli bir lisans altında lisanslanması beklenir ve eLife bunu eLife GitHub deposuna arşivler. eLife, yazılım alıntıları için FAIR ilkelerini uygular. Bununla birlikte, atıf yazılımı Yaşam Bilimleri sektöründe henüz yaygın değildir, bu nedenle schematron (XML'de kalıpların varlığı veya yokluğu hakkında iddialarda bulunmak için kural tabanlı bir doğrulama dili) üretim sürecinde yazılımdan bahseden XML içeriğini aramak için kullanılır - eğer yazar doğru referans vermemişse bunu yapması istenir. Yazılım meta verileri düzenlenmez, ancak bir alıntı oluşturmak için gereken öğelerin mevcut olduğundan emin olmak için kalite kontrolleri yapılır.

IPOL yalnızca özgür yazılımları ve açık kaynak kodlarını kabul eder ve makalelerini ücretsiz bir dokümantasyon lisansı altında yayınlar. Kaynak kodu ayrıntılı olarak gözden geçirilir, algoritmaların matematiksel ayrıntılarına odaklanılır ve uygulamanın makaledeki sözde kod açıklamalarıyla tam olarak eşleşip eşleşmediği kontrol edilir. Kaynak kodu Software Heritage içerisinde arşivlenir ve kendi altyapısında saklanır. Makale ve ilgili kaynak kodu, aynı DOI altında saklanır ve yazılım kaynak kodu, SWHID iç tanımlayıcısı ile sağlanır. Meta veriler otomatik doğrulama yoluyla düzenlenir ve makalenin, yazılımın ve demo editörlerinin tüm yazarlarına ve editörlerine kredi verilir.

Farklı yayıncıların strateji, kapsam ve kaynaklarındaki büyük çeşitlilik göz önüne alındığında, bazıları tarafından daha yüksek düzeyde küratörlüğün uygulanmasına izin verirken, herkes tarafından kolayca uygulanabilen, giriş engeli düşük bir standarda ihtiyaç vardır. Ayrıca yazarlar, dört sütunu desteklemek için kendilerinden ne beklendiğini henüz anlamadıkları için (Bkz. Bölüm 2.1.1), yayıncıların yazarları yazılım kaynak kodunu paylaşmanın gerekliliği konusunda eğitme ve standart bir iş akışını teşvik etme fırsatı vardır.

3.2.3. toplayıcılar

Toplayıcılar, çeşitli topluluklarda bulunabilirliği iyileştirmek için çeşitli kaynaklardan araştırma yazılımı hakkında bilgi toplar, derler, seçer, sunar ve bir araya getirir. Genel olarak, bilgi alanı veri modeli, düğümleri ve kenarları bilinen araştırma ile ilgili varlık türlerine (örneğin yayınlar, veriler, yazılımlar, yazarlar ve kuruluşlar) uyan bir bilimsel bilgi grafiğini tanımlar. Toplanan bilgi türü, hedef veri kaynakları, bunların sonradan işlenmesi ve elde edilen bilgi alanının veri modeli, hedef kullanım durumu uygulamasına bağlıdır. Bilimsel iletişim alanındaki tipik uygulamalar, *keşfedilebilirlik (örneğin kataloglar)* , *kullanım istatistikleri* , *tekrarlanabilirlik* , *araştırma etkisidir (örneğin alıntı dizinleri)*, vb. Bilimsel bilgi grafiği toplayıcılarını iki ana özellik karakterize eder (https://doi.org/10.1007/978-3-030-55814-7_16):

> **Bilgi** toplama Toplayıcılar genellikle dijital veya gerçek dünyadaki nesneleri ve bazı durumlarda dijital nesnenin kendisini tanımlayan meta veri kayıtları toplar, örneğin bilimsel makaleler (OpenAIRE [16] , zbMATH [17] , BASE, Google Akademik, Semantic Scholar [18]), araştırma verileri (örn. Elsevier Data Search, Google Dataset Search).

dahili temsili üzerine eşlenmesiyle ve aynı zamanda tekilleştirme ve tanımlayıcıların oluşturulmasıyla (içsel veya dışsal) ilgili çeşitli işlevsel zorluklar ortaya çıkar. Ayrıca, işlevsel olmayan zorluklar, depolama ve işleme kapasitesi/sürdürülebilirlikle ilgili bir sorundur. Toplayıcılar daha sonra , toplanan verilerin kalitesini ve katma değerini sağlayan, yani madencilik, tarama, çıkarsama, düzenleme, doğrulama vb. yoluyla kendi özel *veri iyileştirme süreçleriyle karakterize edilir.*

Araştırma yazılımını hedefleyen toplayıcı örnekleri olarak şunları bildiriyoruz:

swMATH sadece matematiksel yazılım hakkında kapsamlı bir bilgi veritabanına erişim sağlamakla kalmaz, aynı zamanda yazılım paketlerinin ilgili matematiksel yayınlarla sistematik olarak bağlanmasını da içerir (Bönisch ve diğerleri, 2012b) . Yalnızca belirli tarafların yazılımları kabul edildiğinden, yeni içerik önceden yönetilir. Yazılım değerlendirilirken, meta veriler kaynak kodla uyumlu olup olmadığı kontrol edilir. Meta veriler editörler tarafından toplanır ve yarı otomatik bir süreçle düzenlenir. Atıf ve atıf, yazarlığın referans makaleden çıkarılarak belirlenmesine dayanır. Mümkün olduğunda, Software Heritage'da arşivlenen koda bir bağlantı sağlanır.

OpenAIRE Araştırma Grafiği , içeriği yayınlar, veri kümeleri ve yazılımlardan fon sağlayıcılara, projelere, kuruluşlara, yazarlara ve bilgi kaynaklarına kadar uzanan dünya çapında 12.000 güvenilir bilimsel iletişim kaynağını bir araya getiren açık ve şeffaf bir meta veri koleksiyonudur. Grafik veri modeli, mevcut PID sistemlerine dayanır ve bunları destekler, ancak aynı zamanda URL'leri, harika URL'leri de barındırır [19], ve diğer yerel tanımlayıcılar (kurumsal/tematik depolardan, örneğin ArXiv, EuropePMC, vb.). Araştırma yazılımıyla ilgili olarak, açıklayıcı meta veriler, örneğin kurumsal havuzlar, araştırma yazılımı havuzları (örneğin EGI AppDB, DEO-CODE) gibi bilimsel kaynaklardan toplanır ve veri madenciliği makalesi PDF'lerinin bir sonucu olarak yazılım havuzlarından taranır. bir yazılım havuzuna bir makale tanımlanır. Veri kalitesi, veri kaynaklarına devredilir ve meta veri uyumlaştırma ve bağlam yayılımı yoluyla yönetilir. Yazılım meta veri toplaması için ortogonal ancak anahtar bir aktivite olarak OpenAIRE, DataCite tabanlı bir meta veri profili olan ve Araştırma Yazılımına odaklanan Yazılım Depoları Kılavuzunu [20] oluşturdu.

ScanR , yayın, doktora, patent veya yazılım şeklinde olabilen araştırma ürünlerine özgü bir arama motorudur. ScanR, farklı kayıtlardan kimlikleri kullanarak birbiriyle ilişkili nesneleri birbirine bağlar. Makaleler, GitHub URL'leri taranarak ve bulunursa Software Heritage'a bağlanarak yazılım parçaları için taranır. Şu anda meta veriler düzenlenmemiştir.

3.3. En İyi Uygulamalar ve Açık Sorunlar

Mevcut durumun açıklamasının ardından, TF'nin gelecekteki EOSC'yi destekleyen ideal Araştırma Yazılımı için Akademik Altyapıda uygulanması gerektiğine inandığı en iyi uygulamalar belirlendi.

Bu bölüm, dört sütunun tümü için en iyi uygulamaları kapsar (Bkz . Bölüm 2.1.1) . Her en iyi uygulama için, hangi sütunun ilgili olduğunu, en iyi uygulamanın mevcut durumunun ne olduğunu, hangi boşlukların ele alınması gerektiğini ve boşlukların üstesinden gelmek için ne tür bir eylemin yapılması gerektiğini (araştırma, geliştirme, dağıtım, benimseme) belirtiyoruz. . Üç temsilci grup için geçerli olan endişelerin Bölüm 3.4'te ele alındığını ve her grubun tablolarında tekrar edilmediğini unutmayın.

3.3.1. Arşivler için En İyi Uygulama İlkeleri

Arşivlerle ilgili çalışma grubu, ideal altyapıya yardımcı olmak için ele alınması gereken birkaç boşluk belirledi. Birincisi, tekerleği yeniden icat etmeye gerek olmadığı için, arşiv topluluğu mevcut altyapıları entegre etmek için genel bir mimari üzerinde anlaşmalıdır. Ardından, yazılım arşivleme iş akışı otomasyona ihtiyaç duyar ve geliştirme platformlarıyla entegre edilmelidir. Aşağıdaki tabloda belirtilen iş akışı içerisindeki farklı süreçlerin standardizasyonu ve uyumlu hale getirilmesi ile iş akışının otomasyonu kolaylaştırılacaktır. Ek olarak, yazılım atıfları tanıtılmalı ve metrikler desteklenmelidir. Son olarak, araştırma yazılımı için çeşitli altyapıları birbirine bağlayan ideal mimari, hem açık yazılım hem de açık olmayan yazılımlar için arşivlerin kapsanmasını gerektirir, ve sadece araştırma yazılımlarının değil, tüm yazılımların kaynak kodunun evrensel arşivlenmesini ve referansını sağlama yeteneği. Bu açıkça iki katmanlı bir mimariye yol açar: halka açık tüm yazılım kaynak kodlarını arşivleyen bir alt katman ve akademik topluluğa özel hizmetler sunan çeşitli birbirine bağlı bilimsel havuzlara sahip bir üst katman.

En iyi pratik	sütun	Durum	boşluklar	Oncelikler
Arşivin kapsayıcılığı (ayrıca açık olmayan ve araştırma dışı yazılımlar)	Arşiv	İki katman ortaya çıkıyor, evrensel bir arşiv ve bir bilimsel veri havuzu ağı	ara bağlantı	Gelişim
Yazılım arşivleme iş akışının otomasyonu ve geliştirme platformları ile entegrasyon	Arşiv	Her iki katmanda yapılan ön çalışmalar	Geliştirme platformlarının uzun kuyruğunu koruyun	Gelişim
Arama filtrelerinde desteklenen yazılıma özel kategori	Arşiv Tanım	Farklı uygulamalar mevcut	uyumlaştırma	Benimseme
Yazılım projeleri ve yapıtlar için tanımlayıcılar	Referans	Birden fazla çözüm mevcut, bkz. (Research Data Alliance/FORCE11 Yazılım Kaynak Kodu Tanımlama WG ve diğerleri, 2020)	İçsel ve/veya dışsal PID'lerin ne zaman kullanılacağına ilişkin açık yönergeler	Araştırma uyumlaştırma Benimseme
Yazılım yapılarının sürümlerinin açıkça izlenmesi	Referans Tanım	Mevcut farklı yaklaşımlar (felsefeye bağlı olarak)	Farklı türdeki sürümlerin nasıl ve hangi katmanda izleneceğinin daha iyi anlaşılması	uyumlaştırma
Meta veri sürümlerinin açıkça izlenmesi	Tanım	Bazı arşivler bu bilgileri dahili olarak tutar	Bu bilgiyi harici olarak, en azından katmanlar arasında kullanılabilir hale getirin	Gelişim
Meta veri moderasyonu ve/veya küratörlüğü	Tanım Kredi	Bazı depolar belirli iş akışlarını uygular	Uygulamaların uyumlaştırılması ve iş akışlarının uygulanması	Araştırma
Açık lisans CC0 için meta veri lisanslaması	Tanım	uygulandı	Farkındalık	Benimseme
Yazılım arşivleme yönergeleri	Tanım Kredi	Birden fazla kullanılabilir: aranan hedefe bağlıdır	uyumlaştırma	Araştırma
Destek yazılımı alıntısı	Kredi	Öneriler farklı seviyelerde mevcuttur	Meta veri kalitesi (özellikle atıf) Uygun bibliyografik stiller (bibtex girişleri yalnızca BibLaTeX için son zamanlarda kullanıma sunulmuştur)	Araştırma Gelişim

3.3.2. Yayıncılar için En İyi Uygulama İlkeleri

Bölüm 2.1.1). Bunlar, dergiler üzerinde çalışan alt grup tarafından tanımlanan açık sorunlardan bazılarıdır.

Bu sorunların üstesinden gelmek için aşağıdaki en iyi uygulamalar belirlendi:

En iyi pratik	sütun	Durum	boşluklar	Öncelikler
Gönderim sırasında kaynak koduna referans verme ve arşivleme konusunda yazarlar için destek ve rehberlik	Arşiv Referans Betimlemek Kredi	Bazen ele alınır, ancak maliyetli ve zaman alıcıdır	Eğitim ve öğretim	Gelişim
Kod yazmak için VCS kullanımı	Referans Betimlemek Kredi	Kod barındırma platformları arşiv değildir	Kod barındırma platformlarının farkındalığı	Benimseme
Yayında bir arşive kodun otomatik olarak gönderilmesi	Arşiv Referans Betimlemek Kredi	Henüz kabul edilmedi	Kod barındırma platformları ve arşivler arasındaki farklar ve bağlantılar hakkında farkındalık	Benimseme
Üretilen veya kullanılan yazılımlara atıfta bulunmak ve referans vermek için kullanımı basit araçlar veya yönergeler	Kredi	Gereksinimler anlaşıldı, ancak bebeklik döneminde gelişme	Araç kullanılabilirliği Eğitim	Gelişim
İlişkili makale/veri seti ile bağlantılı olarak yapıtın kalite kontrolü/akran değerlendirmesi	Kredi (inceleyen) Betimlemek	Hesaplamalı Bilimler disiplini ile sınırlı statü	Bunun değerinin tanınması (bu sınırlı destek için kaynak) Çok disiplinli becerilerin eksikliği (yazılım mühendisliği ve bilimsel disiplin)	Benimseme
Kod ve yazılımla ilgili gönderim prosedürü	Kredi Referansı Tanımla	Henüz standart yok	Yönetim eksikliği Eğitim	Gelişim Benimseme

3.3.3. Toplayıcıların En İyi Uygulama İlkeleri

Araştırma yazılımı bilgilerinin toplayıcıları, araştırma yazılımının tanımlanması, tanımlanması ve ifşa edilmesiyle ilgili temel konular tarafından uygun bir açık yazılım altyapısı sağlama çabalarında engellenir. Yazılım oluşturmak ve sürdürmek için tipik olarak kullanılan araçlar, her zaman bilimsel iletişimin ihtiyaçlarını desteklemek için tasarlanmamıştır ve ortak uygulamalar üzerinde anlaşma eksikliği, çeşitli üst veri şemalarının kullanılmasına ve kod havuzlarıyla ilişkili üst veriler arasında yanlış hizalamaya yol açmıştır. ve bu bilimsel amaçlar için gerekli. PID'lerin tutarsız kullanımı, yazılımın benzersiz şekilde tanımlanmasını ve kredinin uygun şekilde atfedilmesini zorlaştırır.

Sonuç olarak yazılım için toplayıcılar oluşturmak sancılı bir süreçtir - araştırma yazılımını neyin oluşturduğunu bile tanımlamak kolay değildir. Bu sorunları gidermek için aşağıdaki en iyi uygulamalar belirlenmiştir:

En iyi pratik	sütun	Durum	boşluklar	Öncelikler

yazılım	Kredi		değildir	
Daha fazla disiplin keşif kaynağı	Arşiv Referans Betimlemek Kredi	Bazıları var (örn. swMATH, BioTools)	Benzer yaklaşımlar diğer topluluklarda da savunulmalıdır Dağıtımı kolaylaştırmak için çözümleri bir "paket" (iş akışları, politikalar, araçlar) olarak sunun	Tanım
Her yazılım girişinin uzun vadeli bir arşive bağlantısı olduğundan emin olun	Arşiv Referans	Embriyonik	evlat edinme eksikliği	Benimseme
PID'lerin kullanımını teşvik edin	Referans	Birkaç yerleşik tanımlayıcı sistemi mevcuttur (Research Data Alliance/FORCE11 Software Source Code Identification WG et al., 2020)	evlat edinme eksikliği	Benimseme
Yazılımın (PID'ler aracılığıyla) diğer araştırma kuruluşlarına (örneğin yayınlar, veri kümeleri, hizmetler, yazarlar) bağlanması	Betimlemek Kredi	Bazı meta veri biçimleri olası çözümler sunar (örneğin DataCite, CodeMeta, Scholix)	Üzerinde anlaşmaya varılan uygulama/anlam bilgisi eksik	Tanımı, onaylanması ve benimsenmesi
CodeMeta'yı tanıtın	Betimlemek Kredi	Kaynak kodu, farklı muhtemelen tutarsız veya çelişkili meta verilerle birkaç yerde bulunabilir.	Ortak meta veri standardı Farklı yazılım meta veri standartları arasındaki yaya geçitleri Yazılım için meta veri çerçeveleri oluşturmak için veri havuzu platformlarıyla etkileşim eksikliği	Benimseme
Meta veri toplama protokolleri (API'ler)	Referans Betimlemek	Araştırma yazılımı meta verilerini ve kodunu paylaşmak için hangi protokollerin kullanılması gerektiği konusunda genel bir anlaşma yoktur.	Toplayıcıların, farklı protokoller (ve formatlar) aracılığıyla yazılımla ilgili meta veri (ve kod) bilgilerini toplamak için kodu desteklemenin (ve sürdürmenin) karmaşıklığıyla yüzleşmesi gerekir.	Tanımı, onaylanması ve benimsenmesi

Bu bölüm, arşivler, dergiler ve toplayıcılar/kataloglar üzerinde çalışan alt grupların sonuçlarına bakarak belirlenen kesişen endişeleri sunar. Her bir öğe için, mevcut çözümler veya gelecekteki çalışmalar yoluyla, onu ele almanın olası yollarını belirtmeye çalışıyoruz.

3.4.1. meta veri

Hat boyunca yazılım eserlerinin doğru bir şekilde tanımlanması gerekir. FUAR'daki dört sütun (Bkz . Bölüm 2.1.1) için esastır . Meta veriler için aşağıdaki ortak gereksinimler tanımlanmıştır:

Ne	Aday çözümler	ARDC	ADİL
Makine tarafından okunabilir, standart format	schema.org , CodeMeta , SPDX	D	F
Yazarlar/katkıda bulunanlar için roller	(Alliez ve diğerleri, 2020a)	C	
Lisans bilgileri (meta veri)	CC0	D	R
Lisans bilgileri (yazılım)	SPDX lisans listesi	D	R
Diğer araştırma çıktılarıyla bağlantı kurma	Diğer araştırma çıktıları için uygun tanımlayıcılarla (örneğin yayınlar için DOI'ler) uygun terimlerin üst veri şemasında (örneğin CodeMeta'daki yayınlar için **referans Yayını) olduğundan emin olun**	D	FR

3.4.2. tanımlayıcılar

Hat boyunca yazılım eserlerinin doğru şekilde tanımlanması gerekir. Dört sütundan (Bkz. Bölüm 2.1.1) R(eferans) ve C(ite) için ve FUAR'da A(erişilebilir) için esastır. Aşağıdaki ortak gereksinimler belirlenmiştir:

Ne	Aday çözümler	ARDC	ADİL
tekrarlanabilirlik için içsel/merkezi olmayan	SWHID	uzaktan kumanda	A
sürümleri ve tüm sw ayrıntılarını destekleyin	çeşitli (içsel ve dışsal)	uzaktan kumanda	A
kalıcı (dışsal tanımlayıcılar için)	çeşitli	uzaktan kumanda	A
standartlaştırılmış	çeşitli (içsel ve dışsal)	uzaktan kumanda	A

İçsel ve dışsal tanımlayıcılara giriş için bu özel blog gönderisine bakın . Daha derin bir analiz için, bkz . Di Cosmo ve ark. (2018); Di Cosmo, Gruenpeter ve Zacchiroli (2020b) .

3.4.3. Kalite ve Küratörlük

Çeşitli derecelerde, yazılım ve/veya *yazılımın kendisiyle ilgili* üst *verilerin kalitesi sorunu* tüm aktörler için ortaya çıkmaktadır. Aşağıdaki kalite yönleri tanımlanmıştır:

Ne	Aday çözümler	ARDC	ADİL
Yazılım kaynak kodunun tekilleştirilmesi	İç tanımlayıcılar (SWHID)	uzaktan kumanda	
Meta verilerin insan küratörlüğü	Çeşitli (bkz. HAL , swMath , ...)	DC	F

intihal tespiti	Manuel inceleme, SWH tarayıcı (gelecek)	C	
Kaynak, yetki kaynağı	Meta verileri kimin, nasıl ve ne zaman ürettiği/küratörlüğünü yaptığı hakkında bilgi tutun; bu, algoritmayı veya yapıtın kendisini tanımlayan makalelere bir referans içerebilir; meta verilerin değişiklik geçmişini tut	DC	R

Tüm paydaşlar/disiplinler, tüm bu iyileştirme ve kalite kontrol seviyelerini uygulamaya muktedir/istekli olmadığından, uygulanan kalite kontrol/kürasyon seviyesinin açık bir göstergesi sağlanmalıdır. Bakılması olası bir fikir, ACM Rozetleme şemasıdır.

3.4.4. Metrikler

Tüm araştırma sonuçlarında olduğu gibi, yazılım için de çeşitli aktörlerin ihtiyaçlarını karşılayacak ölçütler sağlamak faydalı olacaktır. Grup, bu metriklerin *açık, doğrulanabilir ve paylaşılabilir olması gerektiğini kabul eder* .

Bununla birlikte, yazılımla ilgili bilimsel göstergeler , çeşitli nedenlerle, bilimsel literatürdeki belirli bir yazılımın (veya sürümünün) *alıntılarının sayısına basitçe indirgenemez* . Birincisi, yazılımdan alıntı yapmak için hala *yaygın olarak benimsenmiş bir standart yok* : Göründüğü kadar şaşırtıcı, TeX ve Bib(La)TeX bir nesilden fazla araştırmacı için standart araçlar, birinci sınıf olarak yazılımı destekleyen tam teşekküllü bir bibliyografik stil Citizen BibLaTeX kullanıcıları için yalnızca 2020'de kullanıma sunuldu (Di Cosmo, 2020b) . İkincisi, birçok araştırma yazılımının bağlı olduğu önemli yazılım kitaplıkları *doğrudan alıntı yapılmayabilir* (Chawla, 2016; Zhao & Wei, 2017) . Üçüncüsü, araştırma yazılımlarının akademik atıfların basitçe sayılmadığı *akademi dışında* önemli bir etkisi olabilir . Son olarak, bir yazılım parçasının içerdiği katkının *değeri , mutlaka popülerliği* veya yeniden kullanım miktarı ile ilgili değildir [21] .

Tüm bu nedenlerle, geniş çapta tanıtılmadan önce çeşitli ölçütlerin geliştirilmesi ve kaliteleri ve etkileri açısından uygun şekilde değerlendirilmesi gerektiğine inanıyoruz . Bu amaçla, geniş bir uzmanlık yelpazesini bir araya getirmek ve bu metriklerin oluşturulmasından doğrudan etkilenecek araştırma topluluğunun temsilcilerini sohbete dahil etmek gerekli olacaktır.

3.4.5. Yönergeler

Genel olarak eyleme dönüştürülebilir, standartlaştırılmış yönergelere ihtiyaç duyulmaktadır:

- yazılımları kendi kendine arşivleyen araştırmacılar/geliştiriciler için
- bir yayın iş akışında yazılım gönderen araştırmacılar/geliştiriciler için
- yazılım meta verilerini düzenleyen gözden geçirenler/moderatörler için
- yazılımın kendisini değerlendiren gözden geçirenler/moderatörler için
- yayın iş akışlarında yazılımları yöneten yayıncılar için

Geçtiğimiz yıllarda, birkaç kişi, kurum ve çalışma grubu mevcut süreçleri belgeliyor veya yenilerini öneriyor (Alliez ve diğerleri, 2020b; Di Cosmo, 2020b; Di Cosmo, Gruenpeter, Marmol, ve diğerleri, 2020; Gruenpeter & Sadowska , 2018; Katz ve diğerleri, 2020; Smith ve diğerleri, 2016) . Çeşitli genellik ve olgunluk derecelerine sahip bu yaklaşımlar, araştırma topluluğundan gelen geri bildirimleri entegre ederek karşılaştırılmalı ve test edilmelidir.

3.4.6. Araçlar ve İş Akışları

Kılavuzların, onları destekleyecek uygun araçlar olmadan genellikle çok az etkisi vardır. İşte bu araçlarla ilgili birkaç ortak endişe:

- *araştırmacı için katma değer* : araştırmacı tarafından talep edilen herhangi bir eylem, araştırmacının kendisi için anında değer sağlamalıdır.
- *otomasyon* : insan müdahalesi gerektirmeyen görevlerin (örneğin yeni yazılım sürümlerinin arşivlenmesini tetikleme)
- *tekrardan kaçının* : aynı bilgiler farklı formatlarda/yerlerde *girilmemelidir*

tarafından sağlanan CodeMeta oluşturucu aracında alınan yaklaşım budur.

- *endişelerin ayrılması* : veri modeli ve sunum katmanı arasındaki klasik ayrım kavramları geçerlidir. Örneğin , yazılım atıfını destekleyen biblatex-yazılım paketini üretmek için yapılan hazırlık çalışmasına bakın.

4. Önümüzdeki Yol

Bu bölümde, arşivleme, referans, açıklama ve (araştırma) yazılım kaynak kodunun alıntılanmasını destekleyen birbirine bağlı bilimsel altyapıların mimarisi için genel gereksinimleri sunuyoruz. ZORUNLU ve ZORUNLU gibi özel büyük harfli terimler, RFC 2119'da (Bradner, 1997) tanımlanan özel anlama sahiptir ve Sözlük bölümünde geri çağrılır .

Mimarinin üst düzey görünümü Şekil 3'te gösterilmektedir.

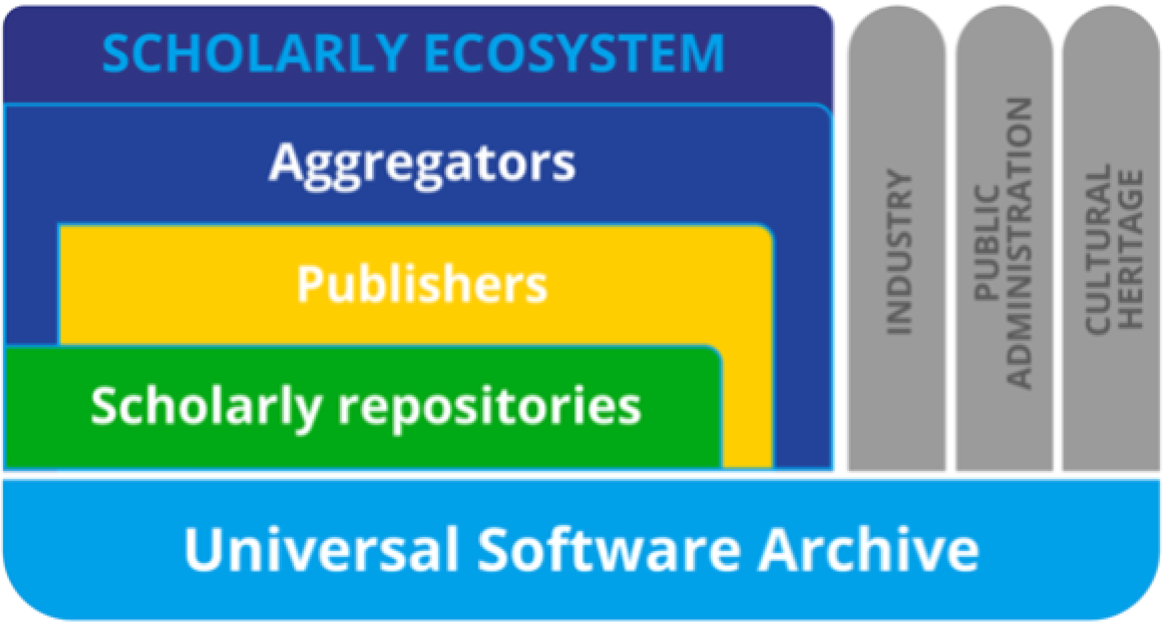


Figure 3. Architecture of interconnected scholarly infrastructures supporting archival, reference, description and citation of (research) software source code.

4.1. Genel Gereksinimler

Temel ARDC işlevlerini uygulamak için gereken katmanlar ve bileşenler.

4.1.1. Arşiv

Amaç

Araştırma sonuçlarının tekrarlanabilirliğini, doğrulanabilirliğini ve yeniden kullanılabilirliğini destekleyin:

- *araştırma yazılımı eserlerinin* uzun vadede korunmasını sağlamak
- *tüm bağımlılıklarının ve ilgili araçların kaynak kodunun* da arşivlenmesini sağlayın (K. Hinsen, 2019)

Bileşenler

1. Katman 0: yazılım kaynak kodu için özel olarak tasarlanmış evrensel arşiv

  - taahhüt eder
  - Salıverme
  - etiketler
  - çatal ve birleştirme işlemleri
  - ilişkili meta verileri (mesaj gönderme vb.)
- birden fazla projede *yazılım kaynağını izleme* yeteneği
- *herhangi bir yazılım eserinin dışa aktarımı* (başka bir kopya yoksa)

2. Katman 1 : bilimsel depolar
   - Aşağıdakilerden bir veya daha fazlasının kimliği belirli kişiler tarafından *açık olarak yatırılması :*
     - ilişkili harici meta verilerle yazılım paketleri
     - evrensel arşivde zaten var olan bir yapıyla ilişkili dışsal meta veriler
   - *kamuya açık olmayan mevduat* ve/veya *ambargo süreleri*
   - harici metadata *düzenleme*
   - (isteğe bağlı) *harici* meta verilerin denetlenmesi
   - yatırılan paketin (olduğu gibi) ve ilgili meta verilerin *indirilmesi*

Bileşenlerin Etkileşimleri

1. Depolar evrensel arşivi beslemelidir ZORUNLU
   - tüm kamuya açık açık mevduatlar evrensel arşive entegre edilmiştir
     - arşive gönderilen yazılım paketleri ve/veya harici meta veriler
       - Bir VCS'den çıkarılan paketler durumunda, evrensel arşivdeki tüm VCS'nin arşivlenmesi tetiklenmelidir.
     - referans tanımlayıcı, onu ortaya çıkaran depoya döndürülür
2. Depolar yerel bir kopya bulundurmalıdır
   - kurumsal, ulusal veya bölgesel politikalar tarafından zorunlu kılınabilir
   - arşivden dışa aktarmadan ziyade depo aracılığıyla daha verimli indirme
3. Evrensel arşiv, mevduatın kaynağını takip etmelidir ZORUNLU
4. Evrensel arşiv, havuza kaynak bilgisi sağlamalıdır ZORUNLU
   - depo mevduatlarının belirsizliğinin giderilmesini desteklemek

4.1.2. Referans

- aşağıdakilerden birinin veya daha fazlasının açık bir şekilde tanımlanmasını sağlayın:
    - isteğe bağlı olarak bağlamında bir yazılım eseri
    - ilişkili meta veriler

Bileşenler

1. İçsel tanımlayıcılar
Yazılım kaynak kodu için özel olarak tasarlanmış, minimum düzeyde güvenilir temel (yalnızca algoritma üzerinde anlaşmaya varılması gerekir)

    - dosyalar, dizinler, taahhütler, sürümler, etiketler ve anlık görüntüler dahil olmak üzere tüm *yazılım yapılarının* merkezi olmayan, bağımsız *olarak tanımlanması*
    - ilişkili yazılım yapılarının merkezi olmayan, bağımsız *doğrulaması :*
        - bir tanımlayıcıyla ilişkili nesneyi idari süreçlerden bağımsız olarak değiştirmenin teknik imkansızlığı (kriptografik olarak güçlü karmalar)
    - *kopyaların* yerleşik tanımlaması
    - geniş çapta kabul görmüş *endüstri standartları ile uyumluluk*

2. Dışsal tanımlayıcılar
Kayıt tabanlı, yapılandırılmış idari gözetim gerektirir

    - *Açık mevduatların* ve bunlarla *ilişkili meta verilerin* havuz kontrollü *tanımlanması*
    - *dijital olarak yerel olmayan bilgilerin* tanımlanması , özellikle:
        - *belirli bir yazılım eserinin* aksine *bir yazılım projesi* kavramı
    - *tanımlayıcıyı değiştirmeden* bir para yatırma ile ilişkili meta verilerin *düzenlenmesi*
    - *bilimsel ekosistemdeki* geleneksel iş akışıyla *uyumluluk*

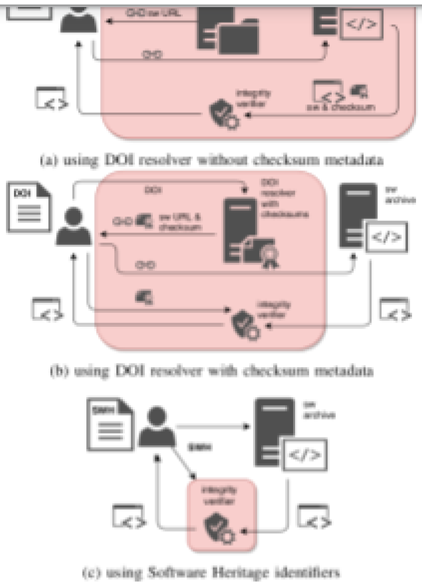


Figure 4. Trusted third parties (shown as rounded red boxes) for software artifact retrieval and verification in three different scenarios.

Güven Modeli

İlgili güven modeli söz konusu olduğunda, içsel ve dışsal tanımlayıcıların çok farklı özelliklere sahip olduğunu fark etmek önemlidir. Dışsal tanımlayıcı sistemleri, tanımlayıcı ile atadığı nesne (meta veri) arasındaki ilişkiyi içeren kayıt defteriyle ilgili işlemleri destekleyen bir altyapı gerektirir. Bir tanımlayıcının belirlenmiş bir nesneyle ilişkisinin kalıcılığı ve doğruluğu, burada Di Cosmo'nun VI . , Gruenpeter ve Zacchiroli (2020a) . İçsel tanımlayıcılar böyle bir altyapıya hiç ihtiyaç duymazlar:tek güvenilir bileşen, tanımlayıcıyı nesnenin kendisinden hesaplamak için kullanılan algoritmadır.

Bu, her ikisi de dijital bir nesne için mevcut olduğunda, içsel bir tanımlayıcıyı asla harici bir tanımlayıcıyla değiştirmemesinin ana nedenlerinden biridir.

Gereksinimler

1. Kamuya açık bir yazılım ürününe yapılan tüm referanslar, nitelikli bir içsel tanımlayıcı içermelidir ZORUNLU; halka açık olmayan bir yazılım eserine yapılan referanslar, içsel bir tanımlayıcı içermelidir ÖNEMLİ

2. Bilimsel bir veri havuzunda açıkça saklanan araştırma yazılımı eserlerine yapılan referanslar, karşılık gelen dışsal tanımlayıcıyı İÇERMEK ZORUNDADIR

3. Yazılım artefaktları olmayan yazılım projelerine yapılan referanslar, nitelikli bir dışsal tanımlayıcıyı İÇERMEK ZORUNDADIR

Tanımlayıcı Sistemlere İlişkin Öneriler

1. Resmi olarak belirlenmiş, açık, tescilli olmayan, sürüm kontrolünden bağımsız içsel tanımlayıcıları kullanın: SWHID'ler önerilir

2. Resmi olarak belirlenmiş, açık, kalıcı, tescilli olmayan harici tanımlayıcıları kullanın

Ortak FORCE11/RDA Yazılım Kaynak Kodu Tanımlama Çalışma Grubu kısa süre önce çeşitli kullanım durumlarını ve yazılım kaynak kodunu tanımlamaya yönelik mevcut yaklaşımları detaylandıran kapsamlı bir rapor yayınladı (Research Data Alliance/FORCE11 Software Source Code Identification WG ve diğerleri, 2020) . DOI'ler, bilimsel yayıncılık dünyasında kritik bir benimsenme düzeyine sahip olduklarından, çeşitli dışsal tanımlayıcı sistemleri arasında belirgin bir avantaja sahiptir. Mevcut yerleşik dış tanımlayıcıların dikkate alınmasını garanti altına almak için kapsayıcı bir yaklaşımın araştırılmasını öneririz.

4.1.3. Betimlemek

Amaç

(Araştırma) yazılım yapılarının *keşfedilebilirliğini* destekleyin

- ○ **içsel:** kaynak kodun kendisinde bulunur
- ○ **dışsal:** bir mevduat, yayın veya toplama işlemi yoluyla oluşturulan

2. Kelime dağarcığı ve ontolojiler
3. Meta verileri oluşturmak, düzenlemek, doğrulamak ve dönüştürmek için araçlar
4. Meta verileri depolamak için kayıtlar

Gereksinimler
Birlikte çalışabilirliği desteklemek için:

1. Meta veriler , *yazılıma uyarlanmış standart* bir sözlük kullanılarak *makine tarafından okunabilir bir biçimde sunulmalıdır ZORUNLU.* CodeMeta (Jones ve diğerleri, 2016) aşağıdaki nedenlerden dolayı iyi bir adaydır:
   - schema.org standardının uzantısı
   - diğer meta veri sözlükleriyle eşlemeye izin vermek için tasarlanmış kapsamlı kelime dağarcığı (CFF ve diğerleri dahil)
   - genişletmek için embrionary topluluk süreci
2. İçsel meta veriler, yazılım geliştirmede tanınan en iyi uygulamalara göre oluşturulmalı ve saklanmalıdır ZORUNLU. Örneğin bkz . https://reuse.software/ ve https://www.tldp.org/HOWTO/Software-Release-Practice-HOWTO/ .
3. Meta veriler ilişkileri desteklemelidir:
   - sürüm oluşturma (aynı yazılımın parçası, yeni sürüm, vb.)
   - diğer araştırma nesneleriyle ilişkiler (makaleler, vb.)
   - diğer tanımlayıcılarla ilişkiler (DOI ve SWHID)
4. Bir yazılım yapıtına özgü bilgiler, içsel meta verilerde OLMALIDIR
   - yazarların/geliştiricilerin bunu sürdürmesini sağlar
   - Bir para yatırma işlemi yapıldığında meta veri girişi çabasını ve kopyalama ve yapıştırma hatalarını azaltır

Öneriler

- Yayıncılar, yayınla ilişkili yazılımın uygun meta verilerle donatıldığından emin OLMALIDIR
- Bilimsel depolar, meta veri iyileştirmeyi desteklemek için gerekli araçları sağlamalıdır.

4.1.4. Alıntı/Kredi

Amaç
(Smith ve diğerleri, 2016) 'da savunulduğu gibi , bireysel katkıların objektif ve kalite değerlendirmesini destekleyen ve mevcut uygulamayı iyileştiren, tatmin edici olmaktan uzak (Howison & Bullard) araştırmada kullanılan yazılım eserlerinin yazarlarına kredi verin . , 2015; Pan ve diğerleri, 2019; Zhao ve Wei, 2017) .

Bileşenler

- mimari
- kodlama
- hata ayıklama
- tasarım
- belgeler
- bakım
- yönetmek
- destek
- test yapmak

2. Yazılım için uyarlanmış bibliyografik alıntı *veri modeli* Bu, tüm ilgili bağlamlarda bir alıntı üretmek için gereken *yazılım üst verilerinin alt kümesidir* . Inria yazılım alıntı çalışma grubundan bir teklif https://gitlab.inria.fr/gt-sw-citation/bibtex-sw-entry/-/blob/master/swentry.org adresinde mevcuttur . Yazılımla ilgili meta verilerin geri kalanı için harici bir kaynağa bağlantı kullanılmalıdır (örneğin, yazarların ilişkisi, adresi ve rollerinin alıntı veri modelinin bir parçası olması gerekmez). Bu bağlantı, bir kayıt defterindeki bir kaydı işaret eden bir PID olabilir (örneğin, kapsamlı DataCite meta veri koleksiyonuna erişim sağlayan bir DOI (DataCite Meta Veri Çalışma Grubu, 2017)
HAL tarafından sağlanana benzer bir kurumsal depo tanımlayıcısı (Di Cosmo, Gruenpeter, Marmol ve diğerleri, 2020) veya astrofizik yazılımı için onlarca yıldır kullanılanlar gibi disipline özgü bir tanımlayıcı (Allen & Schmidt, 2015))
3. Veri modelinin makine *tarafından okunabilir gösterimi Bib(La)TeX veri formatı için Inria* yazılım alıntı çalışma grubundan bir teklif https://gitlab.inria.fr/gt-sw-citation/bibtex-sw-entry adresinde mevcuttur. ///blob/master/swentry.org .

4. Alıntı verilerinin
dizgisi için alıntı stilleri Yazılım için özel olarak tasarlanmış tam teşekküllü bir alıntı stili biblatex -yazılım paketinde CTAN'da mevcuttur ; modüler bir şekilde uygulanır, böylece BibLaTeX kullanıcıları için mevcut *olan diğer yüzlerce alıntı stilinden herhangi birine yazılım alıntıları için destek ekleyebilir* . Bir yazılım alıntı stilinin bu referans uygulaması, mevcut alıntı stillerine yazılım desteği eklemek için bir mihenk taşı olarak kullanılabilir.

5. İntihal tespit mekanizmaları
Araştırma yazılımı değerlendirilen bir öğe haline geldikçe, intihalin ortaya çıkması kaçınılmazdır ve bunu tespit etmek için uygun araçlara ihtiyaç vardır. Benzer araçlar, çok pahalı bir süreç olan *lisans uyumluluğu için endüstride kullanılmaktadır.* Daha hafif araçlar Software Heritage tarafından sağlanabilir. Asgari olarak, yayıncılar tarafından standart manuel kontroller yapılmalıdır.

6. (isteğe bağlı) Uzman akran değerlendirmesi
Birkaç akran değerlendirmesi seviyesi uygulanabilir:

- **en iyisi:** *kabul edilen bir bilimsel yayının ayrılmaz bir parçasını oluşturan yeterli kalitede* araştırma yazılımı sağlayan bir süreç (örneğin , Dagstuhl Artifacts serisinde (DARTS) yayınlanan Bilgisayar Bilimi konferanslarında popüler olan Artefakt Değerlendirme sürecine , ACM Rozetleme şemasına bakın , ve IPOL Dergisi uygulaması)

- **iyi: araştırma yazılımının** *yeni olmasını, geliştirilmesini ve düzgün bir şekilde belgelenmesini ve beklendiği gibi çalışmasını* sağlayan bir süreç (örneğin, Açık Kaynaklı Yazılım Dergisi'nin kriterlerine bakın)

- **orta:** araştırma yazılımının gerçekten kurulabilmesini ve bir araştırma makalesinde yayınlanan sonuçların üretilmesi için kullanılabilmesini sağlayan bir süreç (örneğin, Tekrarlanabilirlik Etiketi ve CodeCheck önerisine bakın)

4.1.5. Evlat Edinmeyi Kolaylaştırmak

Kabulü kolaylaştırmak ve veri kalitesini iyileştirmek için tüm mimaride birkaç yol gösterici ilke geçerlidir:

- *araştırmacı için katma değer* : araştırmacıdan istenen herhangi bir görev, araştırmacının kendisi için anında değer sağlamalıdır.
    - örneğin: bibliyografik girdilerin, müfredatın veya form doldurmanın otomatik olarak oluşturulması
    - örneğin: muhtemelen depolar arasında toplanan şeffaf ve doğrulanabilir metrikler (indirmeler, görüntülemeler)
- *üst veri doğrulama* : kredi veya değerlendirme için kullanılabilecek üst veri, insan doğrulamasından geçmelidir
- *tekrardan kaçının* : aynı bilgiler farklı biçimlerde/yerlerde girilmemelidir ve özellikle:
    - kullanıcıların makine tarafından okunabilen meta verilerden çıkarılabilecek bilgileri manuel olarak girmesi gerekmemelidir
- *eyleme geçirilebilir yönergeler* : yönergeler, takip edilmesi kolay ve uygulanabilir adımlar sağlamalıdır
- *altyapılar arasında otomasyon ve entegrasyonu tercih edin*

4.2. Katılımcı Altyapılar için Örneklik Kriterleri

Finansman kurumları ve kamu kurumları, açık akademik altyapıları finanse etmek için örnek olma kriterlerine bakıyorlar (Bilder ve diğerleri, 2015) ve bilimsel altyapı ile ilgili finansman kararlarının nasıl alınması gerektiğine dair kılavuzlar sunmaya başladılar. Açık Erişim alanında, COAR ve SPARC (COAR & SPARC, 2019) , GO-FAIR veya fransız NFSO (Fransız Açık Bilim Ulusal Komitesi, 2019) gibi kuruluşlar arasında paylaşılan genel ilkeler bulunabilir .

Bu kriterlerin çoğu , bilimsel *çıktıyla ilişkili* üst verilerin *açıklığı* ve *kullanılabilirliği* ile ilgilidir: bunlar bizim ortamımızla açıkça ilgilidir ve bunları bu bölümde sunulan mimarinin tasarımına dahil ettik. Diğer kriterler, altyapıların açıklığı, sürdürülebilirliği, şeffaflığı ve yönetimi ile ilgilidir.

Burada, WG'nin araştırma için ilgi çekici yazılımları ele alan bilimsel altyapılar için özellikle önemli olduğuna inandığı kriterleri özetliyoruz.

açıklık

- meta verilere standart bir biçimde ve bir CC0 lisansı altında erişilebilir olmalıdır
- meta verilere ve verilere erişim, standart protokoller kullanılarak ve tanımlama olmaksızın açık bir API aracılığıyla mümkün olmalıdır.
- toplu meta veriler "mümkün olduğunca açık, gerektiği kadar kapalı" olmalıdır (örneğin, GDPR düzenlemelerine uymak için)
- altyapılar, kararlı, mevcut açık kaynaklı yazılım yapı taşlarından inşa edilmeli ve altyapının tüm yazılımları bir açık kaynak lisansı altında mevcut olmalıdır.
- iletişim ve veri alışverişi, veri biçimleri ve protokolleri için açık standartlar kullanır
- altyapı, mülk edinme riskinden kaçınmak için kar amacı gütmeyen bir kuruluş tarafından barındırılmalı ve çalıştırılmalıdır.

Yönetim

- yönetişim organlarının net tanımı
- yönetişim organlarının üyelerinin seçimine ilişkin prosedürler açıkça ve kamuya açık olarak belirtilir
- katılım prosedürleri açıkça ve kamuya açık olarak belirtilmiştir

Sürdürülebilirlik

- altyapının veya platformun genel işleyişi, tek seferlik projelerin finansmanına dayanmamaktadır.
- hizmetin uzun vadeli kullanılabilirliği için bir plan mevcuttur ve kamuya açıklanır
- hizmet ömrünün ötesinde verilere ve meta verilere süreklilik sağlayabilecek bir çıkış stratejisi

şeffaflık

4.2.1. Yerinde Inovasyon

Gelecekte yeni yenilikçi hizmetler ortaya çıkabilir ve burada önerilen mimarinin amaçlarından biri de bu yeniliklerin kolayca dahil edilmesini mümkün kılmaktır.

Bu, önerilen mimarinin çeşitli bileşenler arasında iletişim ve bilgi alışverişi için *açık standartlara* ve *açık formatlara* dayanması gerçeğiyle mümkün olmaktadır. Özellikle, *meta veri* alışverişi için ortak bir *açık standart* (CodeMeta), içsel tanımlayıcılar (SWHID) için ortak bir *açık standart* , *açık API'lerin* kullanımını önermek ve altyapıların kaynak kodunun kullanılabilirliğini aşağıdaki şekilde benimsemeyi öneriyoruz. *açık kaynak* .

Bu şekilde, arşivleme, yayınlama veya birleştirme için yeni yenilikçi hizmetler ortaya çıkarsa, mimarinin diğer mevcut bileşenleri ile sorunsuz bir şekilde birlikte çalışabilecekler.

4.3. Olası İş Akışları

Bu bölümde, ilgili adımların görselleştirilmesine izin veren karşılık gelen dizi diyagramları ile birlikte tanımlanmış referans senaryoları hakkında daha fazla ayrıntı sunuyoruz.

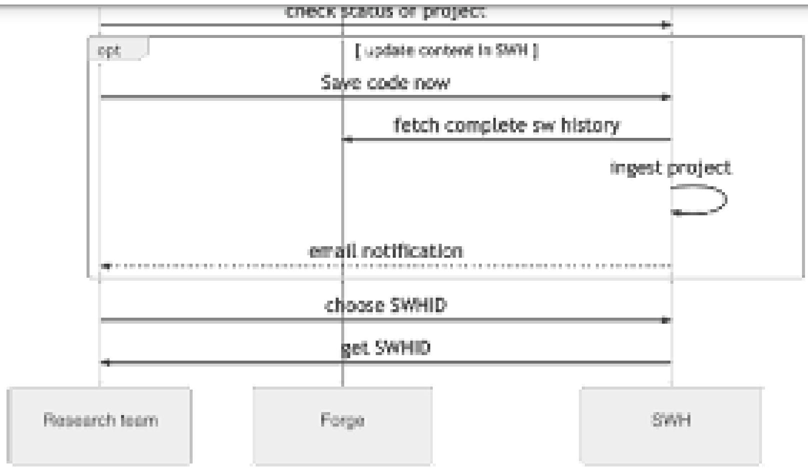
Aşağıda, aşağıdaki rolleri ayırt ediyoruz:

- **araştırma ekibi:** yazılım kaynak kodunu geliştirir veya kullanıcıları; ARDC ile ilgileniyor
- **forge:** kaynak kodun ortaklaşa geliştirilmesi için kullanılan kod barındırma platformu
- **yayıncı:** akademik yayın kuruluşu (makale ve/veya kaynak kodu incelemesi)
- **bilimsel arşiv:** bir araştırma kurumu tarafından yürütülen arşiv (örn. HAL, Zenodo)
- **katalog/toplayıcı:** (örn. OpenAIRE, swMATH, Scanr)
- **SWH:** Software Heritage evrensel kaynak kodu arşivi
- **veri kaynağı** : toplayıcılar için girdi kaynağı (yayınlar, meta veriler)

4.3.1. Kendi Kendine Arşivleme

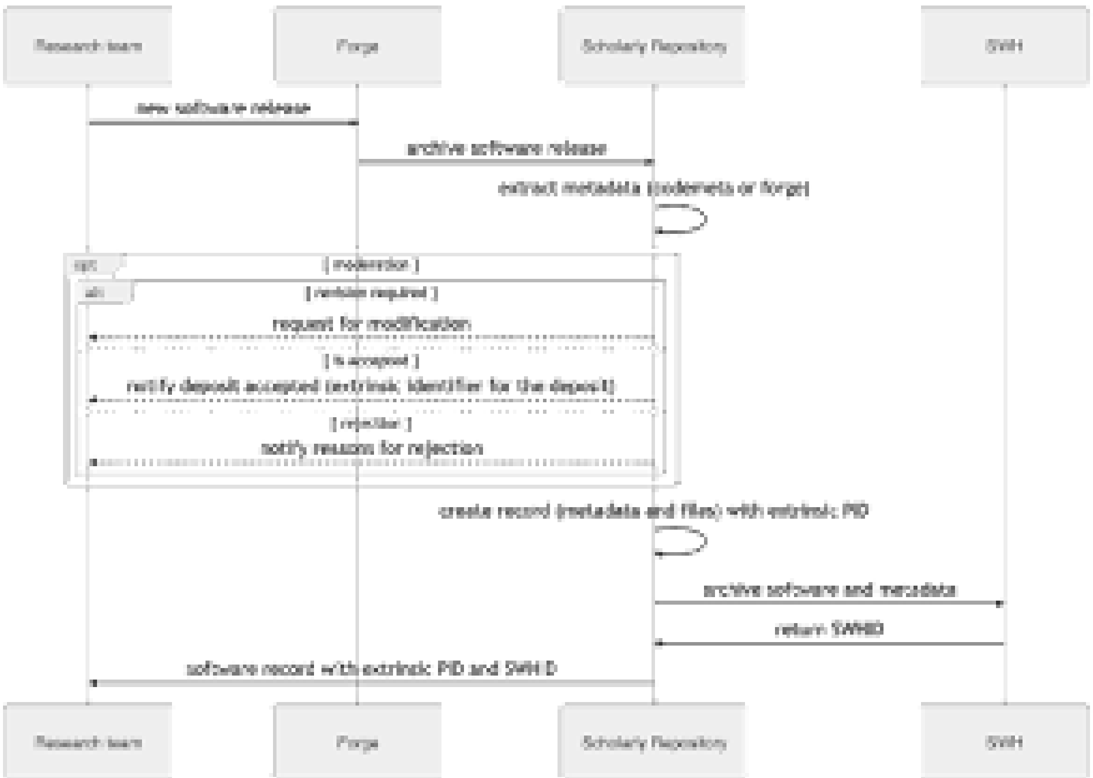
Bir araştırma ekibi, bir yazılım eserini kendi kendine arşivler. Arşiv bölümünde zaten açıklandığı gibi , bu doğrudan SWH'de veya kaynak kodun *meta verilerinin belirli bir düzeyde kalite incelemesini/küratörlüğünü* destekleyebilen veya desteklemeyen bir Akademik Depo aracılığıyla yapılabilir.

SWH'deki Forges Arşivi (Manuel ve Otomatik)
Yazılımlarını herkesin erişebileceği kod barındırma platformlarını kullanarak geliştiren ekipler için en basit varyant. Araştırma ekibi, yazılım projesinin (en son sürümünün) halihazırda SWH'de arşivlenip arşivlenmediğini kontrol eder, gerekirse arşivlenmesini ister ve ardından ilgili SWHID'yi alır. Arşivleme talebi, SWH API aracılığıyla bir yayın sürecinin veya sürekli entegrasyon sürecinin bir parçası olarak manuel olarak veya otomatik olarak gönderilebilir. Bu , Yazılım Mirası ile Kaynak Kodu Arşivleme ve Referans Verme, ICMS, pp. 362–373, Springer, 2020'de ayrıntılı olarak açıklanan iş akışına karşılık gelir .



Yukarıdaki şemada açıklanan iş akışı, yalnızca belirli bir araştırma ekibi tarafından geliştirilen belirli bir yazılım projesine değil, aynı zamanda bir kod barındırma platformunda *halka açık herhangi bir yazılım projesine de uygulanabilir.*

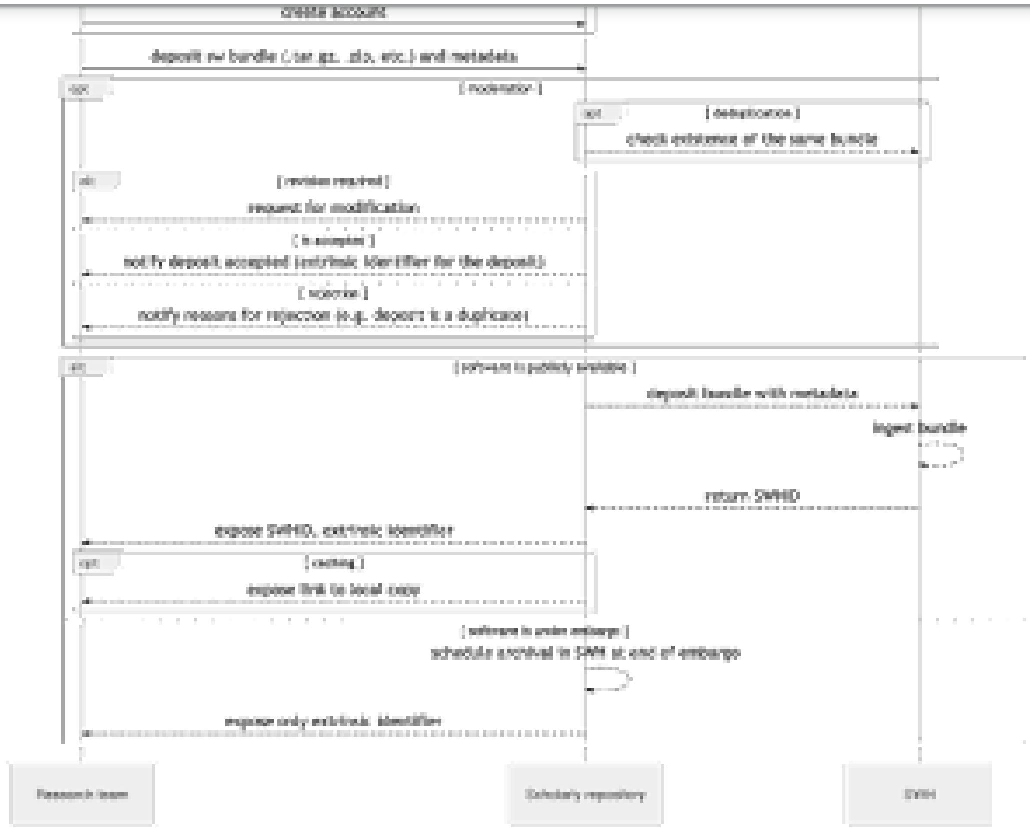
Yeni Çıkanların Scholarly Repository ve SWH'ye otomatik olarak yatırılması (Manuel ve Otomatik)
Araştırma ekibinin açık bir yayın süreci varsa ve belirlenmiş bir bilimsel veri havuzu seçmişse , güncellenmiş içeriğin yeni bir denetimini tetikleyebilecek bir bilimsel veri havuzuna yatırma sürecini otomatikleştirmek mümkün olabilir (örneğin, ArXiv veya HAL'de bir araştırma makalesinin yeni sürümlerinin depolanması).



SWH'deki denetleme ve arşivleme dışında yukarıdaki iş akışı, GitHub forge'ı Zenodo'ya bağlamak için uygulandı.

Bir Akademik Depo Aracılığıyla Para Yatırma Paketi
Bir araştırma ekibi , *bir yazılım paketini* (.tar.gz, .zip, vb.) ve ilişkili *meta verileri* bilimsel bir havuza açıkça yerleştirir. Bilimsel veri havuzu , mevduatın belirli bir *kalitesini sağlamak için bir denetleme* mekanizması uygulayabilir (tekilleştirme, ekip üyelerinin bağlantıları, meta verilerin tutarlılığı, vb.). Kabul edildikten sonra paket ve meta veriler, hemen veya isteğe bağlı ambargo süresinin sonunda SWH'de arşivlenir.

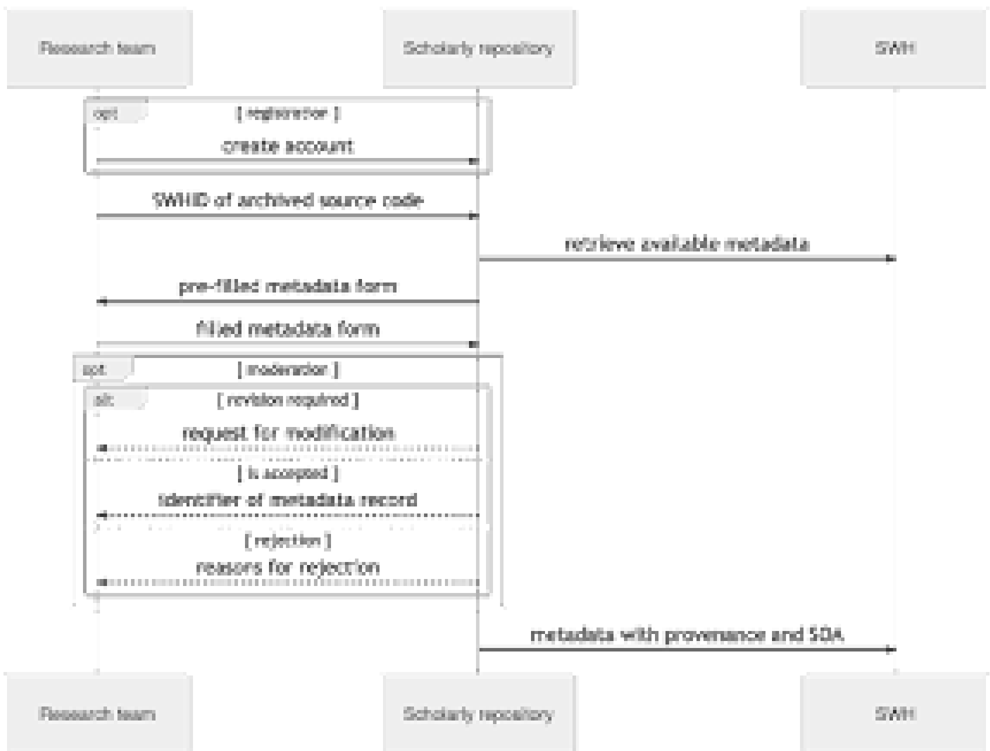
Aşağıdaki sıra şeması , Fransız ulusal açık erişim deposu HAL tarafından halihazırda uygulanmış (isteğe bağlı tekilleştirme hariç) adımları temsil etmekte ve özel bir araştırma makalesinde ayrıntılı olarak anlatılmaktadır (Di Cosmo, Gruenpeter, Marmol, ve diğerleri, 2020) .



Denetleme adımının kaldırılmasıyla elde edilen basitleştirilmiş iş akışı, şu anda InvenioRDM'yi SWH ile birbirine bağlamak için değerlendiriliyor.

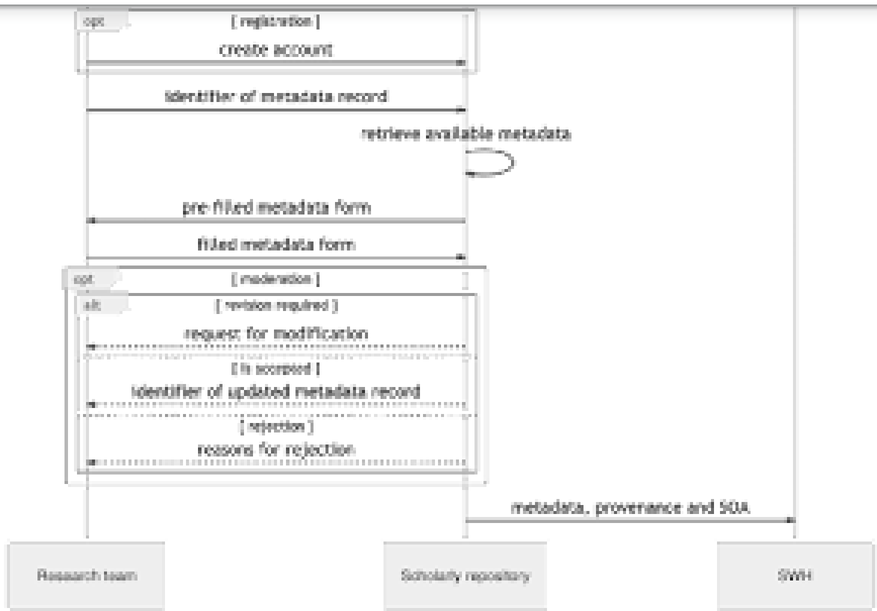
Halihazırda Arşivlenmiş Yazılımları Bir Akademik Depoya Kaydetme

Bir araştırma ekibinin, SWH'de zaten arşivlenmiş bir eseri bilimsel bir havuzda kaydetmesi gerekebilir . Çalışmanın tekrarlanmasını önlemek için, meta veri yatırma formunu önceden doldurmak için kaynak kodunda belirtilen dosyalarda bulunan makine tarafından okunabilir meta veriler kullanılmalıdır. Bu iş akışı şu anda HAL açık erişim deposunda uygulanmaktadır.



Bir Akademik Veri Havuzunda Mevcut Meta Verileri Güncelleme

Bir araştırma ekibinin ayrıca, halihazırda arşivlenmiş bir yazılım eseri hakkında bilimsel bir havuzda kayıtlı meta verileri güncellemesi gerekebilir. Bu iş akışı şu anda HAL açık erişim deposunda uygulanmaktadır. Zenodo ayrıca yükleyicilerin meta verileri güncellemesine izin verir ve gelecekte iyileştirme özelliklerini öngörür.



Üst veri kaydının önceki sürümleri, kimin neyi, ne zaman ve neden değiştirdiğine ilişkin bilgilerle birlikte kaydedilmeli ve mevcut olmalıdır.

4.3.2. İlişkili Kaynak Kodlu Bilimsel Yayın

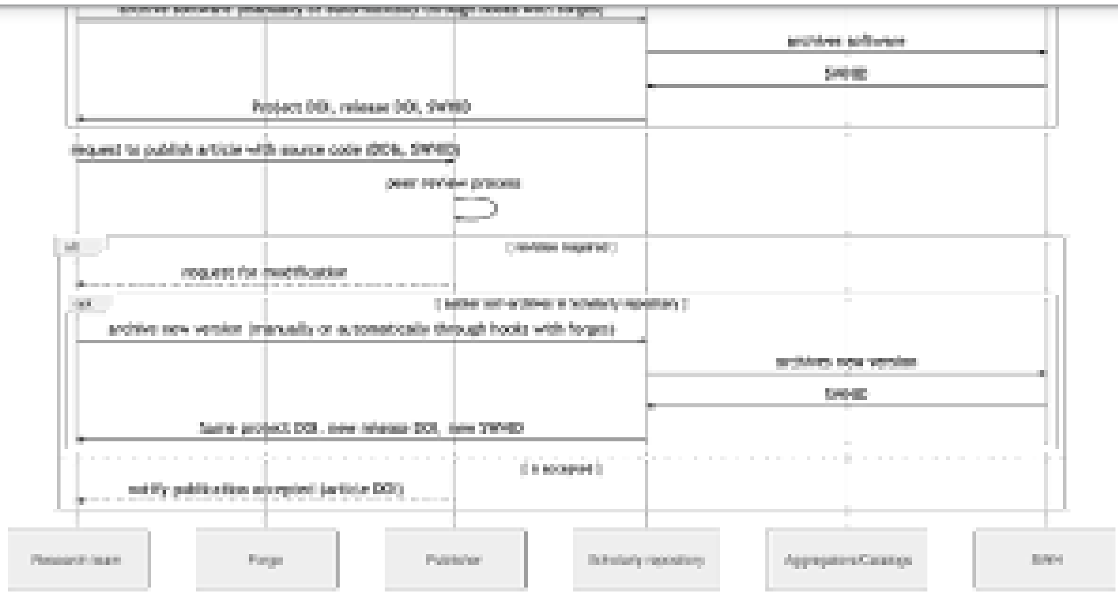
Bu senaryoda, bir araştırma ekibi, ilgili yazılım kaynak koduyla bir makaleyi bir yayıncıya gönderir. Yayıncının uyguladığı inceleme/küratörlük düzeyine bağlı olarak çeşitli varyantlar mümkündür. İlgili vakalardan birkaçını detaylandırıyoruz.

Kaynak Kodu Tamamen Yazar Tarafında İşlenmiştir
Yayıncı, kaynak kodun ve/veya ilişkili meta verilerin gözden geçirilmesi/düzenlenmesi için herhangi bir özel iş akışı uygulamaz. Bu durumda araştırma ekibi, yukarıda açıklanan kendi kendine arşivleme iş akışlarından herhangi birini izleyerek ilgili kaynak kodunu kendi kendine arşivler ve yayının son sürümüne uygun tanımlayıcıları ekler.

Yayıncı, Public Forge'da Barındırılan Herkese Açık Kaynak Kodun İncelemesini Uyguladı
Yayıncı, kaynak kodun ve/veya ilişkili meta verilerin belirli bir düzeyde kalite incelemesini/küratörlüğünü destekler. İş akışı, her bir yayıncının seçtiği iş akışına bağlı olarak değişebilen inceleme sürecini detaylandırmaz, ancak inceleme için değiştirilmemiş kaynak koduna erişimin gerekli olduğunu ve herkese açık kaynak kodu için bunun, yazarın yazarlarının kimliğini ifade ettiğini belirtiriz. kod mutlaka açığa çıkar. Bu, kaynak kodun çifte anonim incelemesinin uygulanmasını imkansız hale getirir (yani yazarlar hakemleri tanımıyor ve hakemler yazarları tanımıyor). Sonuç olarak,Ya yazarların hakemler tarafından tam olarak bilindiği inceleme süreçleri (dergilerde en yaygın seçenek, örneğin IPOL'ün ne yaptığına bakın ; açık inceleme veya tek anonim inceleme olabilir) veya iki aşamalı bir inceleme süreci bulunabilir. ilk aşama makalenin kendisine odaklanmıştır (bu çift anonim hale getirilebilir) ve ikinci aşama, makale kabul edildikten sonra, yazılımın yazarlarının gözden geçirenler tarafından bilindiği [23] ve ücretsiz olarak etkileşimde bulunduğu yazılıma odaklanır. (Bilgisayar bilimlerinde onlarca prestijli konferans tarafından oluşturulan ve ACM Rozetleme sisteminde önerilen Artefakt Değerlendirme Komiteleri tarafından yapılan şey budur ; Dagstuhl DARTS serisi bu süreçlerin sonuçlarının bir seçimini yayınlar).

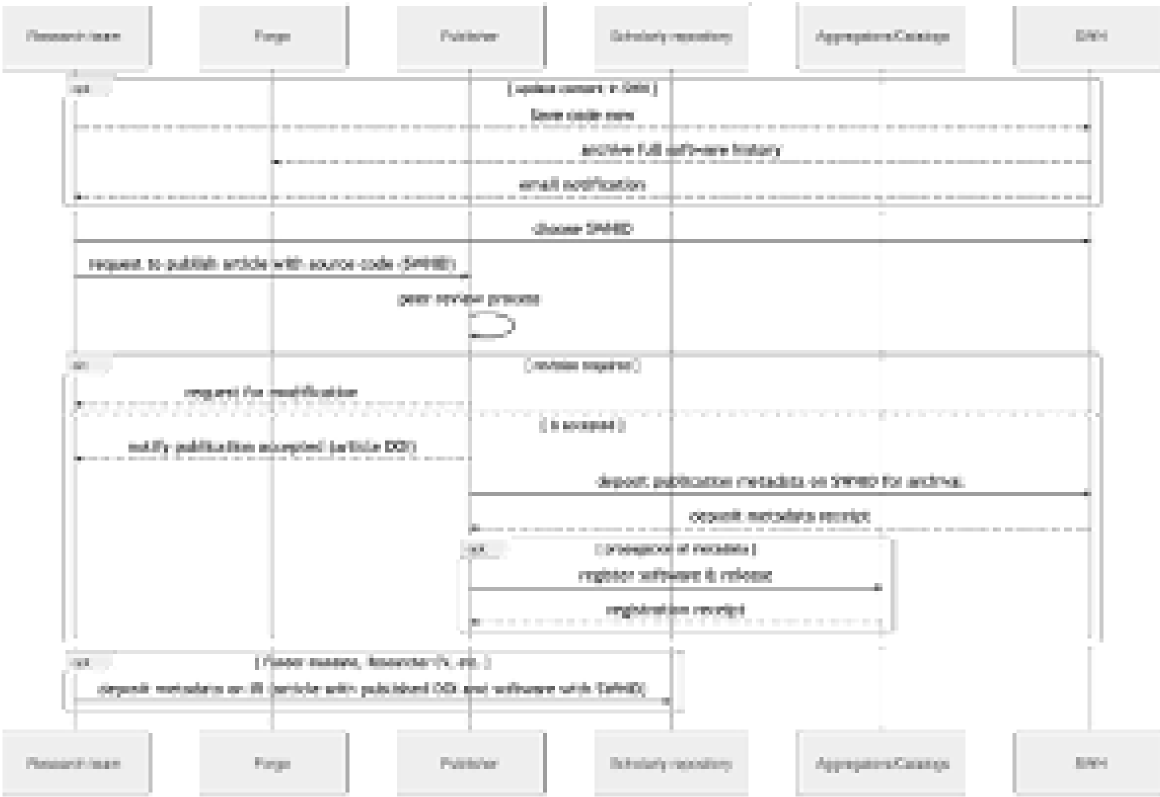
Yayıncı, yazılım ürününü içeren yayının bir Akademik Depoya (SWH'ye yayılan) veya doğrudan SWH'ye ve isteğe bağlı olarak ilgili toplayıcılara/kataloglara kabulüyle ilgili meta verileri yayar; genel olarak bunun yazılım eseriyle ilgili dışsal meta veriler olduğuna dikkat edin. Yayıncı, yayınlanan makaleyle ilişkili kaynak kodunun arşivlenmesinde de rol oynayabilir. Birkaç varyant aşağıda gösterilmiştir.

Varyant 1.a. Bir Akademik Depoda yazarın kendi arşivini oluşturma

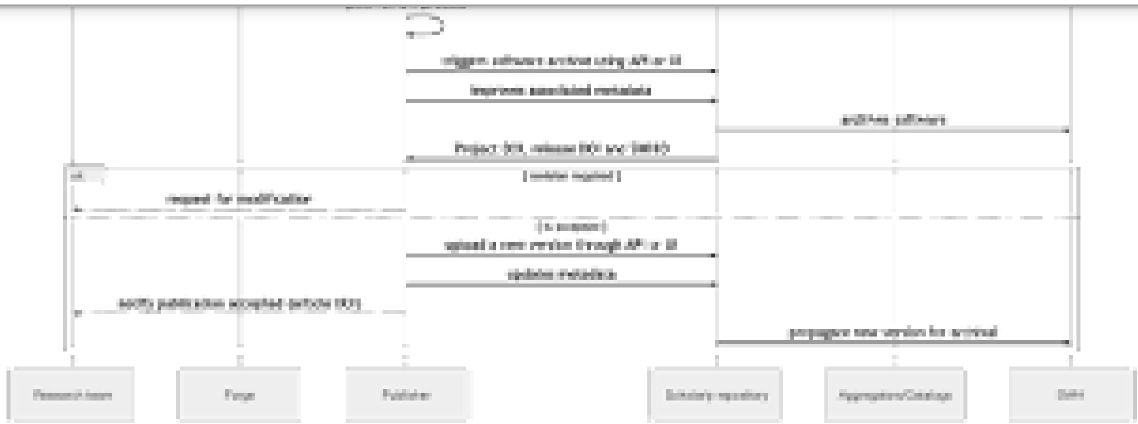


Bu iş akışı, tüm revizyonlar boyunca aynı PID'yi koruyarak inceleme sürecini kolaylaştıran Proje DOI'lerinin kullanımı olan bilimsel yayında yaygın bir durumu göstermektedir.

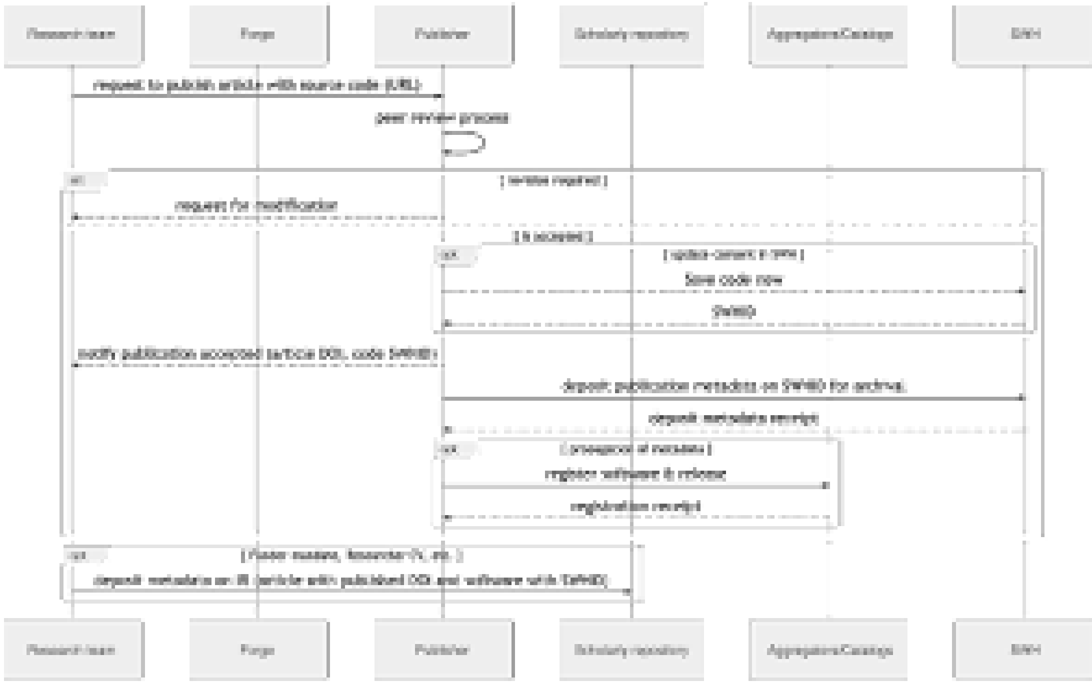
Varyant 1.b . Yazar kendi kendine arşivler doğrudan SWH'de



Varyant 2.a . Bilimsel veri havuzundaki yayıncı arşivleri



Varyant 2.b . Yayıncı arşivleri doğrudan SWH'de

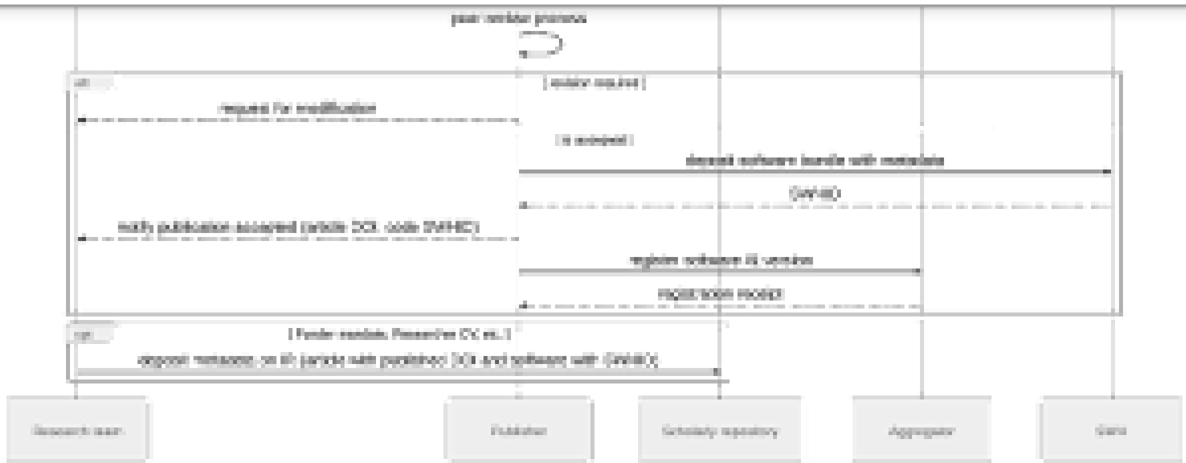


Yayıncı tüm arşivleme adımlarını ele aldığından, bu varyant daha az yazar müdahalesi gerektirir.

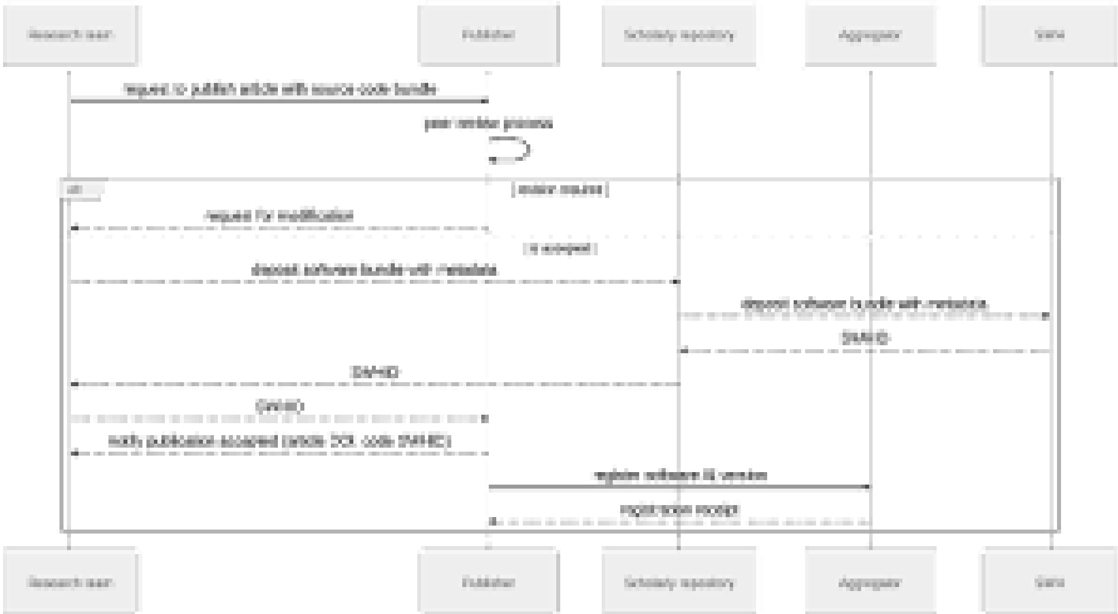
Yayıncı, Bir Paket Olarak Gönderilen Kaynak Kodunda İnceleme Uygular
Yazılım bir paket olarak gönderilirse, ya yayıncı tarafından ya da bilimsel bir veri havuzu aracılığıyla SWH yatırma API'sı aracılığıyla arşivlenmesi gerekir.

Varyant 1. Yayıncı arşivleri doğrudan SWH'de

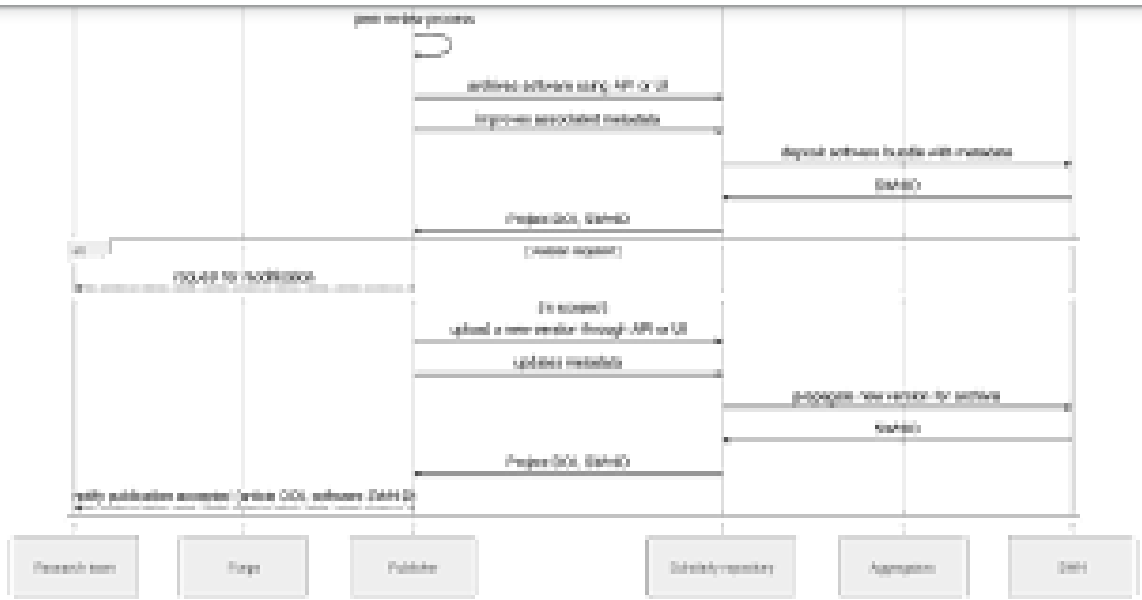
Bu, şu anda IPOL tarafından uygulanan iş akışıdır, bkz . https://www.softwareheritage.org/2020/06/11/ipol-and-swh/ .



Varyant 2. Yayıncı , yazardan bilimsel veri havuzunda arşivlemesini ister



Varyant 3. Bilimsel veri havuzundaki yayıncı arşivleri



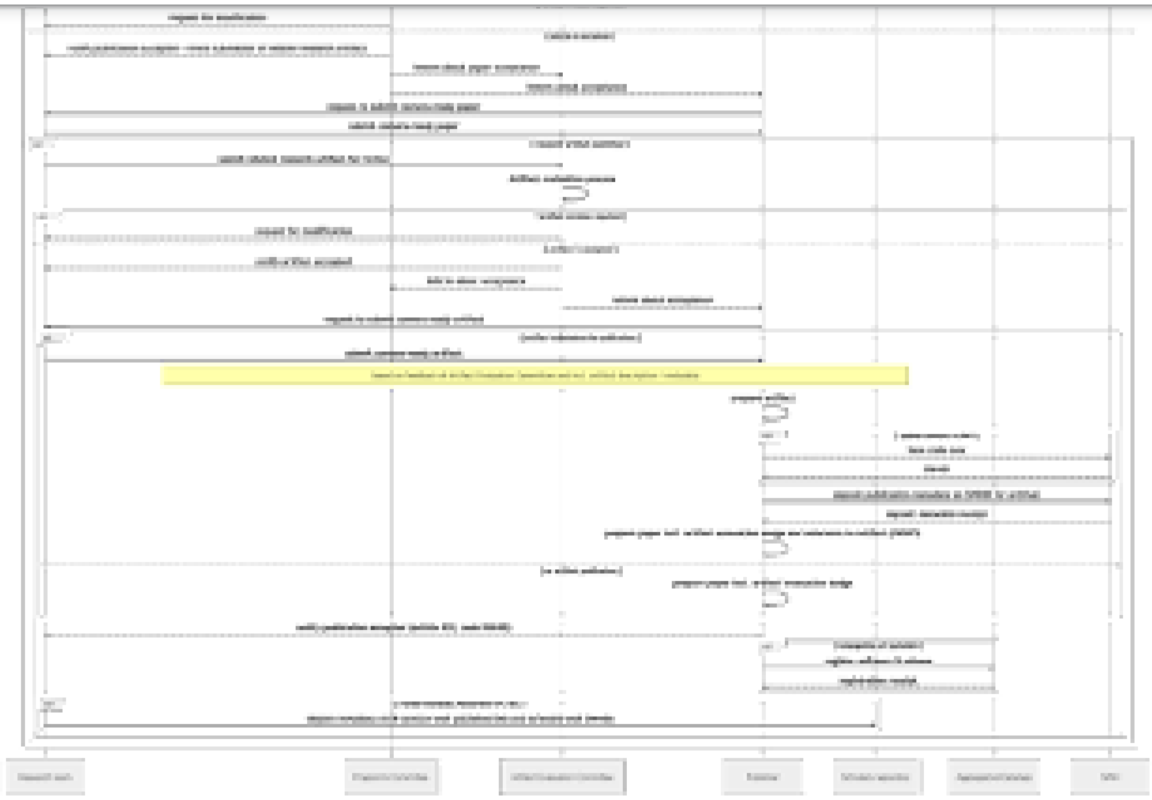
Dryad ve Zenodo [24] tarafından geliştirilen bir prototipte yukarıdakine benzer bir iş akışı uygulanmaktadır .

Eser Değerlendirme Komiteleri ile Konferanslar
Bilgisayar biliminde, diğer birçok disiplinde olandan farklı olarak, araştırma topluluğu dergilerden çok konferans bildirileri yayınlamaya alışkındır. Bunun, program komitesinin düzenli olarak değişmesinden, çoğu durumda konferansın kimliğinin bildirilerin yayıncısıyla çok az ilgisi olmasına kadar uzanan bir dizi ilginç sonucu vardır. Burada, bir Artefakt Değerlendirme Komitesi (AEC; (Childers ve diğerleri, 2016) [25] yerleştiren birçok prestijli bilgisayar bilimi konferansında 2011'den beri yaygın olarak benimsenen iş akışına dayalı bir iş akışıyla ilgileniyoruz .

Aşağıda gösterilen iş akışı, birkaç açıdan öncekilerden farklıdır:

- konferansın Program Komitesi belirli bir yayıncıya bağlı değildir
- *eserin değerlendirilmesi, sadece kabul edilen makaleler için* Program Komitesinden *farklı olarak* özel bir AEC tarafından gerçekleştirilir ; sonuç olarak, eserler söz konusu olduğunda artık anonim incelemeye gerek yoktur ve AEC yazarlarla serbestçe etkileşim kurabilir;
- yayıncı, yapıt değerlendirme sürecine hiç dahil değildir. Yalnızca eser olumlu olarak değerlendirildikten sonra iki eylemi gerçekleştirmek için devreye girer.
  - kalıcı bir tanımlayıcı elde ederek yapıtı kaydedin
  - kaydedilen esere isteğe bağlı bir bağlantıyla makaleye bir rozet ekleyin

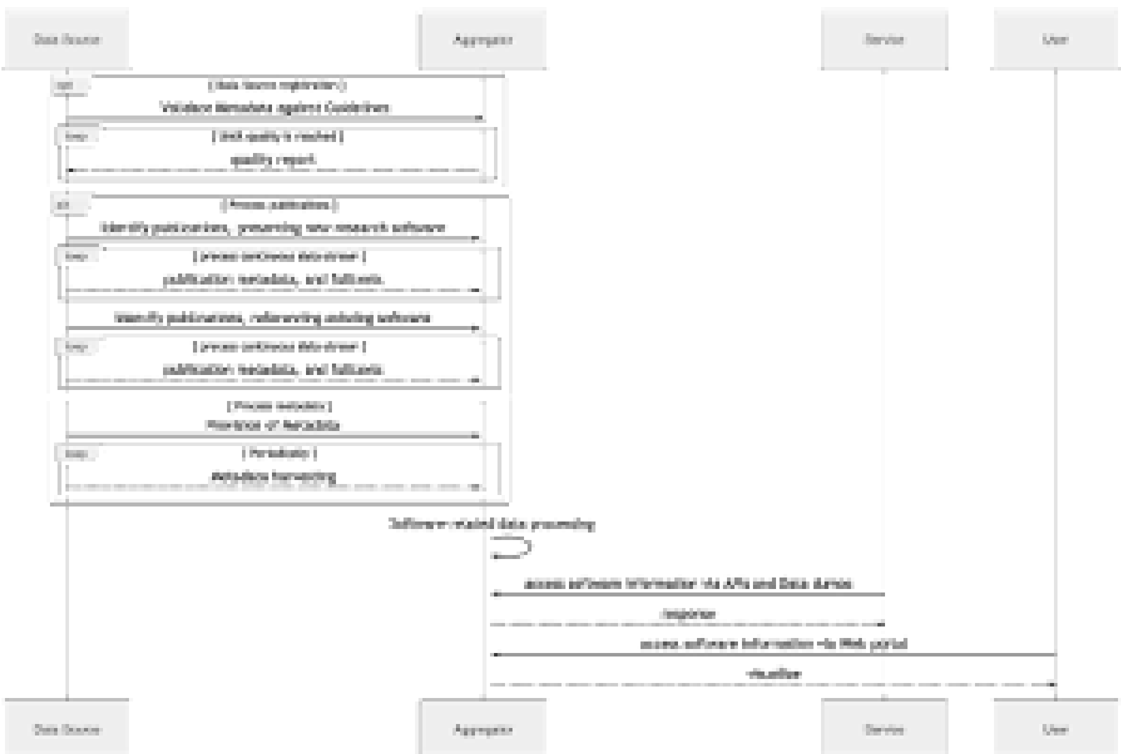
Şu anda, eser değerlendirme iş akışı konferans işleme yazılımında yerel olarak desteklenmediğinden ve eserin arşivlenmesi genellikle yazarlara bırakıldığından, bu süreç için destek tam olarak tatmin edici değildir. Bazı konferanslar, SWH [26] 'da veya yayıncının kendi dijital kütüphanesinde arşivleme yapılmasını önerir, ancak iş akışında gösterildiği gibi, arşivlemeyi sağlama sorumluluğunu yayıncının üstlenmesinin daha iyi olacağına inanıyoruz. Yayıncılık sistemlerinde kullanılan yazılımlara, farklı kalite inceleme seviyeleri ile eser değerlendirme ve arşivleme için uygun destek eklenmelidir. Bu, AEC sürecinin veya ondan esinlenen süreçlerin daha geniş anlamda konferanslarda ve dergilerde alınmasına izin verecektir.



4.3.3. toplayıcılar

Toplayıcıların iki gruba veya modele uyduğu tespit edilmektedir: İtme modeli ve Çekme modeli. İlki, *Veri Kaynaklarının* proaktif olarak yazılımla ilgili meta verileri ve yayınları *Toplayıcılara gönderdiği bir modeldir* ; ikincisi, *Toplayıcıların* yazılımla ilgili meta verileri ve yayınları doğrudan *Veri Kaynaklarından aldığı bir modeldir.*. Modelden bağımsız olarak, toplayıcıların bilimsel yayıncılık iş akışlarındaki ana rolü, her zaman bilgiyi toplamak, işlemek (uyumlaştırma, veri tekilleştirme, zenginleştirme vb.) Toplu bilgiler daha sonra son kullanıcılara daha iyi hizmetler sunmak için aynı Veri Kaynakları (örn. Arşivler) tarafından tekrar kullanılabilir.

çekme modeli
Toplayıcılar, Veri Kaynaklarından yazılımla ilgili meta verileri ve/veya yayınları proaktif olarak toplar. Yayınlar söz konusu olduğunda işlenir, yazılımla ilgili bilgiler çıkarılır ve benzersiz bir şekilde tanımlanır. Meta veri durumunda, doğrulanır ve dahili bilgi grafiğiyle birleştirilir.



İtme modeli

Toplayıcılar, API'lerin diğer Hizmetlere (örn. Arşivler) yeni yayınları (meta veriler dahil) toplamaya gerek kalmadan doğrudan göndermelerine olanak tanır.



OpenAIRE, bu Push Modeline ve ayrıca Kodlu Belgelere [27] sığabilir . DataCite, geleneksel bir toplayıcı olmasa da, Push modeline uygun ve tek bir toplayıcı olarak görülebilir; Bunun nedeni, Akademik Depoların yazılımla ilgili meta verileri doğrudan DataCite'a göndermesi ve toplu meta verilerin API'leri ve Kullanıcı Arayüzü aracılığıyla erişilebilir olmasıdır.

5. Öneriler

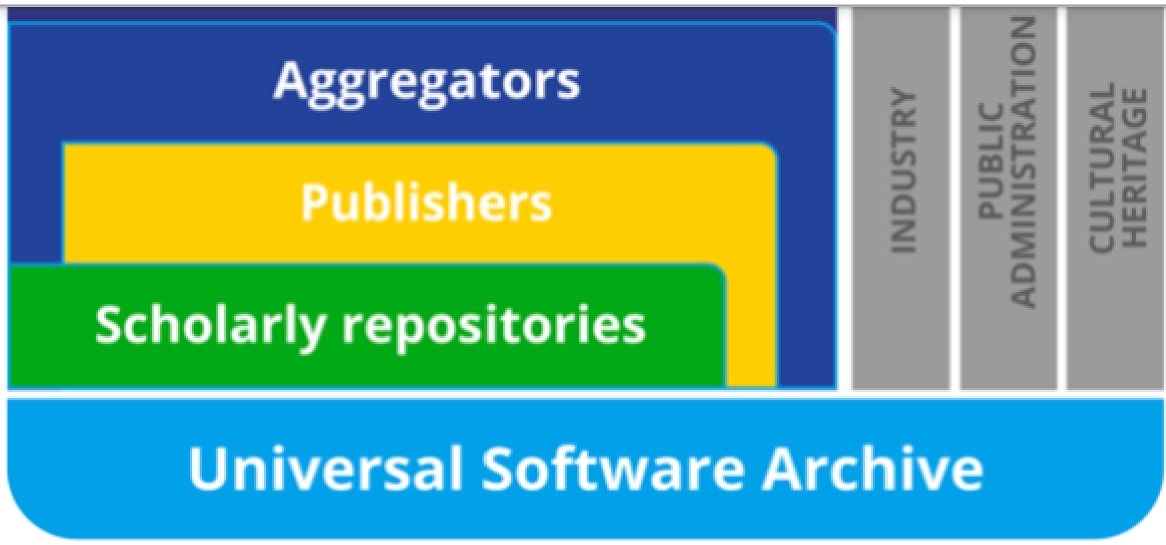


Figure 5. Architecture of interconnected scholarly infrastructures supporting archival, reference, description and citation of (research) software source code.

Bu raporun girişinde hatırlatıldığı gibi, araştırma yazılımı için bağlantılı bilimsel altyapıların uygun bir mimarisini oluşturmak önemli bir girişimdir ve *standartlar, araçlar, altyapılar, eğitim, erişim* ve *yayıncılık topluluğuyla katılım gerektirir* . Aynı zamanda hem *geliştirme, iletişim* ve *sosyal yardım çabaları* hem de *operasyonel maliyetler için uygun finansmana ihtiyacı var* . Bu bölümde, mevcut ihtiyaçların ve son teknolojinin analizinden ve geleceğin mimarisinin tasarımından ortaya çıkan temel önerileri özetliyoruz .

5.1. Araçların, Standartların ve Yönergelerin Geliştirilmesine Finansman Sağlanması

Aşağıdaki öneriler dizisi , kısa vadede somut kalkınma projelerine dönüştürülmesi gereken *eyleme geçirilebilir maddeleri içermektedir.*

5.1.1. Etkileşimler

Bölüm 4.1.1'de tartışıldığı gibi , evrensel bir yazılım arşivi ile bilimsel arşivlerin evrensel arşivi beslemesi için, bilimsel veri havuzları arasında dikey bir bağlantının sağlanması önemlidir (bkz. Şekil 5). Bu, uygun adaptörlerin geliştirilmesi için mühendislik ve finansman gerektirir. Dağıtım 2 yıllık bir zaman diliminde öngörülmektedir. Aynı zamanda, yayıncıların yazılım arşivleme için kendi çözümlerini uygulamalarından ve yazılımların korunmasını ve alıntılanmasını sağlamak için bilimsel havuzlara veya evrensel bir yazılım arşivine güvenmelerinden kaçınılmalıdır. Yazılım kaynak kodunu bilimsel havuzlara ve evrensel bir yazılım arşivine yerleştirme veya arşivleme eylemi dahil Yayıncılık iş akışında , dahili süreçlerin uyarlanmasını gerektirir ve 2 yıllık bir zaman diliminde uygulanabilir. Yazılım kaynak kodu arşivleme ve referans iş akışını otomatikleştirmeye yönelik araçların geliştirilmesi, mühendislik ve finansman gerektirir ve 4 yıllık bir zaman diliminde uygulanabilir. Ek olarak, araştırma sonuçlarının tekrarlanabilirliğini ve doğrulanabilirliğini desteklemek için evrensel bir yazılım arşivinde arşivlenen kaynak kodun yanı sıra küratörlü üst verilerin arşivlenmesi sağlanmalıdır. Bu nedenle, küratörlü meta veri arşivleme iş akışını otomatikleştirmek için araçların mühendisliği ve finansmanı gereklidir ve 4 yıllık bir zaman diliminde uygulanması öngörülmektedir.

5.1.2. Yazılım Hakkında **Meta Veriler**

Aşağıdaki öneriler dizisi, yazılımla ilgili meta verilerle ilgilidir. İlk olarak, yazılımla ilgili tüm meta veriler CC0 olarak lisanslanmalıdır. İkincisi, mimarinin farklı bileşenleri arasındaki tüm meta veri alışverişi CodeMeta sözlüğüne dayanmalıdır. AB temsilcileri CodeMeta topluluğuna katılmalı ve istikrarlı, uzun vadeli bir yönetişim kurulmasına yardımcı olmalıdır. Ayrıca, CodeMeta sözlüğü kullanılarak meta verileri tüketmek ve ortaya çıkarmak için gerektiğinde ek dönüştürücüler ve bağdaştırıcılar geliştirilmelidir. Bu, mühendislik ve finansman gerektirir ve 2 yıllık bir zaman diliminde devreye alınabilir. Üçüncüsü, meta veriler, özellikle lisans, tanımlayıcılar, havuzlar, yazarlar ve yazılım geliştirmeyi destekleyen fon sağlayıcılar dahil olmak üzere, bilimsel dünyadaki yazılım kaynak koduyla ilgili tüm bilgileri içermelidir. (örneğin AB veya ulusal hibeler). Son olarak, Yayıncılar tarafından alıntılar için kullanılan genel formatların (örn. JATS) yazılım alıntılarıyla ilgili tüm üst veri öğelerini düzgün bir şekilde desteklediğinden ve yazılım alıntıları için

5.1.3. tanımlayıcılar

Yeni tanımlayıcı sistemlerinin oluşturulması istenmeyen bir durumdur ve bunun yerine yeniden kullanım teşvik edilmelidir. Aşağıdaki tanımlayıcı listesinin kullanımının genelleştirilmesi uygundur:

1. Herkese açık yazılım kaynak kodu için **SWHID içsel tanımlayıcıları.**
2. Bilimsel bir veri havuzunda açıkça saklanan araştırma kaynak kodu için dışsal tanımlayıcılar .
3. Yazılım projeleri için dış tanımlayıcılar (ayrıntılar için Bölüm 4.1.2'ye bakın).

DOI'ler, bilimsel yayıncılık dünyasında kritik bir benimsenme düzeyine sahip olduklarından, çeşitli dışsal tanımlayıcı sistemleri arasında belirgin bir avantaja sahiptir. Mevcut yerleşik dış tanımlayıcıların dikkate alınmasını garanti altına almak için kapsayıcı bir yaklaşımın araştırılmasını öneririz.

5.1.4. Kredi

Kabulü kolaylaştırmak için, araştırma yazılımı kaynak kodu için uygun alıntılar üretebilecek araçların geliştirilmesi ve aynı yaklaşımı destekleyecek mevcut referans yönetim araçlarının geliştirilmesi arzu edilmektedir. Spesifik olarak bu , temel alınan veri modelinin yazılım kaynak kodunun tüm özelliklerini (Alliez et al . ., 2020a; Di Cosmo, 2020a; Katz ve diğerleri, 2020; Research Data Alliance/FORCE11 Yazılım Kaynak Kodu Tanımlama WG ve diğerleri, 2020) .Ek olarak, yayıncılar, ortak bir alıntı stilleri kümesi üzerinde anlaşmaya vararak, yazılıma özel olarak nasıl alıntı yapılacağına ilişkin yönergeler üretmeli ve genel olarak araştırma topluluğuyla birlikte doğrulamalıdır. Ayrıca, yayıncılar, eldeki araştırma için yazılmamış herhangi bir koda ilişkin tüm bağlantıların/bahsetmelerin, ilgili tüm gerekli meta veriler dahil olmak üzere, kaynakçada uygun referanslar olarak ele alındığından emin olmalıdır. Çeşitli aktörlerin ihtiyaçlarını karşılayan ölçütler araştırılmalı ve bunları paylaşmak ve yeniden kullanmak için ortak standartlar üzerinde anlaşmaya varılmalıdır. Bölüm 3.4.4'te detaylandırıldığı gibi , bu metrikler açık, doğrulanabilir ve paylaşılabilir olmalıdır. Bibliyografik göstergelerin araştırma yayıncılığı dünyasında sahip olduğu olumsuz etkiyi araştırma yazılımı dünyasında yeniden üretmekten kaçınmak için bu metrikler basit sayısal göstergelere indirgenmemelidir. Geniş bir uzmanlık yelpazesini bir araya getirmek ve bu metriklerin oluşturulmasından doğrudan etkilenecek araştırma topluluğunun temsilcilerini sohbete dahil etmek gereklidir . Yayıncılar, Bölüm 4.1.4'te belirtildiği gibi, meslektaş inceleme sürecinin kendi alanlarına en uygun değerlendirme düzeyiyle yazılım kaynak kodunu da kapsadığından emin olmalıdır. (nokta 6) ve moderasyon ve küratörlük protokolleri için bir dizi ortak kılavuz geliştirin. AEC'ler tarafından uygulananlar gibi daha karmaşık gözden geçirme düzeylerinin benimsenmesini desteklemek ve kolaylaştırmak için bir dizi standart araç ve iş akışının geliştirilmesi finanse edilmelidir.

5.1.5. Politika/Yönergeler

Araştırma yazılımı için bağlantılı bilimsel altyapıların uygun bir mimarisini oluşturmak, yazılımın birinci sınıf bir vatandaş olarak ele alınmasını artırmak için kılavuzlara ihtiyaç duyar. Spesifik olarak, araştırmacılar için Açık Bilim yönergeleri, yazılımların bilimsel havuzlarda ve Software Heritage tarafından sağlanan evrensel yazılım arşivinde depolanmasını açıkça önermelidir. Aynı zamanda, Açık Bilim yönergeleri, yazılım geliştirmeye (VCS'ler, sürekli entegrasyon vb. dahil) yönelik modern yaklaşımların varlığı hakkında farkındalık yaratmalı ve uygun olduğunda bunların kullanımını teşvik etmelidir. Son olarak, tüm yayıncılar kaynak kodun ve verilerin önemi konusunda bilinçlendirilmeli ve makale yayını ile birlikte bu eserlerin yayınlanmasını ve arşivlenmesini zorunlu hale getirmelidir.

5.1.6. Evlat Edinmeyi Kolaylaştırmak

Bölüm 4.1.5'te detaylandırıldığı gibi , benimsenmeyi kolaylaştırmak ve toplanan (meta)verilerin kalitesini artırmak için özel dikkat gösterilmesi önemlidir. Bu amaçla, araştırmacıların katılımı büyük önem taşımaktadır ve araçlar ve kılavuzlar geliştirirken, *araştırmacılar için anında bir katma değer olduğundan emin olunmalıdır* (örneğin, bibliyografik girdilerin, müfredatın veya form doldurucuların otomatik olarak oluşturulması, mevcut prosedürlerin basitleştirilmesi). , vb.) ve *araştırmacıların bilgileri bir kereden fazla manuel olarak girmeleri* veya makine tarafından okunabilir meta verilerden çıkarılabilen bilgileri girmeleri gerekmemelidir. *Özel bir hassas faktör, yazarlara kredi vermek veya araştırmacıları değerlendirmek* için kullanılabilecek bilgilerin kalitesidir., bu nedenle bu amaçla kullanılabilecek meta veriler *insan küratörlüğünden* geçmelidir (Alliez ve diğerleri, 2020a) .

5.2. EOSC için Daha Geniş Politika Önerileri

Çeşitli aktörler için teknik geliştirme, teknik standartlar ve genel yönergelerin yanı sıra, EOSC'nin genel mimarinin araştırma topluluğunun ihtiyaçlarını en iyi şekilde karşılayacak şekilde inşa edilmesini sağlamada önemli bir rolü olduğuna inanıyoruz. , COAR ve SPARC (COAR & SPARC, 2019) , GO-FAIR veya fransız NFSO (Fransız Açık Bilim Ulusal Komitesi, 2019) gibi kuruluşlar arasında Açık Erişim altyapıları hakkında paylaşılan genel ilkelerle aynı ruhla .

Aşağıdaki öneriler dizisi, EOSC'nin temel altyapıların açıklığını, şeffaflığını, iyi yönetimini ve sürdürülebilirliğini sağlaması için somut eyleme geçirilebilir adımlar sunmaktadır.

İlk olarak, EOSC, Bölüm 4.2'de ayrıntıları verilen açıklık, iyi yönetişim ve şeffaflık ilkelerini içeren katılımcı altyapılar için bir dizi mükemmellik kriteri hazırlamalıdır . Açıklık ilkesine uyum için aşağıdaki kriterler listesine dahil edilmelidir:

- m etverileri standart bir biçimde ve bir CC0 lisansı altında erişilebilir olmalıdır
- meta verilere ve verilere erişim, standart protokoller kullanılarak ve tanımlama olmaksızın açık bir API aracılığıyla mümkün olmalıdır.
- toplu bir meta veri "mümkün olduğunca açık, gerektiği kadar kapalı" olmalıdır (ör. GDPR düzenlemelerine uymak için)
- altyapılar, kararlı, mevcut açık kaynaklı yazılım yapı taşlarından inşa edilmeli ve altyapının tüm yazılımları bir açık kaynak lisansı altında mevcut olmalıdır.
- iletişim ve veri alışverişi, veri formatları ve protokolleri için açık standartlar kullanır
- altyapı, mülk edinme riskinden kaçınmak için kar amacı gütmeyen bir kuruluş tarafından barındırılmalı ve çalıştırılmalıdır [28]

Ayrıca, şeffaflık ve iyi yönetişim ilkelerine uyum için aşağıdaki mükemmellik kriterleri listesi dahil edilmiştir:

- yönetişim organlarının net tanımı
- yönetişim organlarının üyelerinin seçimine ilişkin prosedürler açıkça ve kamuya açık olarak belirtilir
- katılım prosedürleri açıkça ve kamuya açık olarak belirtilmiştir
- kullanım koşulları açıkça ve kamuya açık olarak belirtilmiştir
- finansman kaynakları açıkça ve kamuya açık olarak belirtilir

İkincisi, EOSC, uzun vadeli sürdürülebilirliklerini sağlamak için kilit altyapılarla aktif olarak ilgilenmeli ve temel Açık Kaynak projelerinin tarihinde yıllar içinde tanımlanan ortak tuzaklardan kaçınarak stratejik evrimlerinde yer almalıdır (Eghbal, 2016) ve araştırma verileri için altyapılarla ilgili kaygılarla aynı olmayan (Rosenthal ve diğerleri, 2014) . Bu, birkaç yönü içerir:

- *Teknik* : arşivlerin veri koruma için ***standart uygulamaları benimsemesini sağlayın.***
- *Mali/Kurumsal* : önemli altyapılar için ***uzun vadeli bir finansman planına*** ve/veya tasfiye ve göç planına katkıda bulunun. Özellikle, *altyapıların işletimini finanse etmek için proje parasına güvenmeyin.*
- *Organizasyonel* : mimarinin temel bileşenlerinin *kar amacı gütmeyen bir altyapı olarak çalıştırıldığından emin olun* ***ve yönetişime*** aktif olarak katılın.

5.3. Daha Uzun Vadeli Perspektifler

Daha uzun vadeli bir bakış açısıyla, aşağıdaki hedeflerin yol haritasında net bir şekilde yer alması gerektiğine ve 4-7 yıllık ufukta bunları ele almak için araştırma ve geliştirme çabalarının harcanması gerektiğine inanıyoruz.

5.3.1. İleri Teknoloji Geliştirme

Önemli olan, SWH tarafından sağlanan evrensel kaynak kodu arşivine ek olarak verimli ve açık bir intihal tespit teknolojisinin geliştirilmesidir. Bu, arşivlerin, yayıncıların ve toplayıcıların, geleneksel yayınlarda olduğu gibi, ancak makaleler için gerekli olan ancak açık kaynakta bulunmayan kapalı veri kümeleri ve ticari sözleşmelerin sınırlamaları olmaksızın, neredeyse kopyaları tespit etmesine ve sahtekarlıktan kaçınmasına olanak tanır. yazılım dünyası . Benzer teknoloji, yüksek maliyetler ve farklı ana hedeflerle endüstride kullanılmaktadır ve yeniden hedeflenmesi ve bilimsel dünyaya uyarlanması gerekmektedir. Software Heritage tarafından bir araya getirilen küresel yazılım kaynak kodu külliyatı, bu görev için önemli bir etkinleştiricidir.
Ayrıca, yazılım dışı projelerin filtrelenmesine izin veren verimli ve açık spam filtreleme araçları oluşturmak, mevduatların insan tarafından denetlenmesini zorunlu kılmayan bilimsel havuzları ve arşivleri korur ve bunu yapanlar için moderatörlerin işini kolaylaştırır .

etmelidir.

Araştırma sonuçlarının tam olarak yeniden üretilebilirliğini geliştirmek için, belirli bir yürütülebilir dosyanın veya tam bir yazılım sisteminin ve iş akışlarının makaleler, veriler ve yazılımlar arasında uygun entegrasyon ile tekrar güvenilir bir şekilde yürütülebilmesini sağlamak için mekanizmalar, teknoloji ve araçlar araştırılmalıdır. Bu, araştırılması ve değerlendirilmesi gereken çok çeşitli araçlar ve yaklaşımlar içeren karmaşık bir konudur.

5.3.2. Politika

Tüm araştırma çıktılarının "mümkün olduğunca açık, gerektiği kadar kapalı" olması gerektiğine dair genel ifadeye katılmakla birlikte, Açık Bilimin potansiyeline tam olarak ulaşmak için ***tüm araştırma yazılımlarının bir Açık Kaynak lisansı altında varsayılandır ve bu varsayılan uygulamadan tüm sapmalar uygun şekilde motive edilmelidir*** . Bu maddeyi gelecekteki tüm araştırma finansman programlarına dahil etmenizi öneririz.

6. Ekler

6.1. Sözlük

Terim	***Tanım***
Toplayıcı	Bu raporun hedeflediği ***operasyonel altyapıların*** ağaç tipolojilerinden biri . Öncelikli olarak keşfedilebilirliğini artırmak ve muhtemelen küratörlük, soyutlama ve sınıflandırma ve bağlama gibi süreçler yoluyla bu bilgilere değer katmak amacıyla çeşitli kaynaklardan dijital içerik hakkında bilgi toplayan herhangi bir hizmet. Bilimsel katalogları ve dizinleri içeren bu hizmet sınıfı, genellikle toplu içeriğin açıklamasına erişim sağlayan bir arama motoru sağlar ve başka yerde arşivlenmiş sürümlerine bağlantılar sağlayabilir. Bu hizmetler genel olabilir veya disipliner, coğrafi veya kurumsal bir kapsama sahip olabilir.
Mimari	*Mimari* terimi genellikle bir sistem oluşturmak için gereken bileşenlerin üst düzey tasarımını ve bunların ilişkilerini ifade eder. Bu raporda , bilimsel dünyadaki ARDC ihtiyaçlarını karşılayabilecek operasyonel altyapıların üst düzey organizasyonunu ve ilişkisini belirtmek için *altyapıların bilimsel mimarisi terimini kullanıyoruz.*
Arşiv	Bu raporun hedeflediği ***operasyonel altyapıların*** ağaç tipolojilerinden biri . Topladığı dijital içeriğin uzun vadeli korunması birincil hedeflerinden biri olan herhangi bir hizmet. Bu, akademik dünyada kurumsal arşivlerden disiplin arşivlerine kadar geniş bir hizmet yelpazesini ve

	bir yayınla ilişkili yazılım ürününün kalitesini değerlendiren, genellikle program komitesinden ayrı olan bir gözden geçirenler paneli. Daha fazla ayrıntı için [https://www.artifact-eval.org/ adresine](https://www.artifact-eval.org/) bakın.
ARDC	*Arşiv* , *Referans* , *Tanımla* ve *Alıntı Yap* anlamına gelen bir kısaltma , yani araştırma dünyasında yazılım kaynak kodu yönetimi için bilimsel altyapıların desteklemesi gereken dört sütun
KodMeta	Depolar ve kuruluşlar arasında yazılım meta verilerinin değişimini standartlaştırmak için kullanılabilecek bir kavram sözlüğü (CodeMeta şeması) geliştirmek için çağrılan bir proje.
Dış tanımlayıcılar	Tanımlayıcı ile belirlenmiş nesne arasındaki yazışmayı tutmak için bir kayda dayanan tanımlayıcı sistemleri. Bilimsel dünyada çok iyi bilinen örnekler ARK, Handle ve DOI'dir.
ADİL	Başta veri kümeleri olmak üzere dijital varlıkların Bulunabilirlik, Erişilebilirlik, Birlikte Çalışabilirlik ve Yeniden Kullanımını teşvik etmek için geliştirilmiş bir dizi ilke
altyapı	***Operasyonel Altyapıya*** Bakın
İç tanımlayıcılar	Yazışmayı sürdürmek için bir kayda ihtiyaç duymadan, belirlenmiş nesnenin kendisinden hesaplanabilen tanımlayıcılar. Dijital çağdan önce iyi bilinen içsel tanımlayıcı sistemleri müzik notasyonu ve kimyasal notasyondur. Yazılım kaynak kodu için geçerli standart SWHID'dir.
JAT'LAR	Dergi Makale Etiket Paketi
ZORUNLU	RFC 2119'da olduğu gibi kullanılır: Bu kelime veya "GEREKLİ" veya "GEREKLİ" terimleri, tanımın spesifikasyonun mutlak bir gerekliliği olduğu anlamına gelir.
Operasyonel Altyapı	Bir organizasyonun işleyişi için ihtiyaç duyulan konuşlandırılmış tesisler, ekipman, hizmetler, politikalar, prosedürler ve insan kaynaklarından oluşan karmaşık bir sistem.
PID	PID, "Kalıcı Tanımlayıcı" anlamına gelir; genellikle dış tanımlayıcı sistemlerini (örn. DOI, Ark, Handle) belirtmek için kullanılan bir terimdir ve tanımlayıcılar ile belirlenmiş nesneler arasındaki ilişkiyi sürdürmek için kurumsal desteğin kurulduğu bir terimdir.
Yayımcı	Bu raporun hedeflediği ***operasyonel altyapıların*** ağaç tipolojilerinden biri . *Bir yayın üretmek ve yayımını, tanıtımını ve arşivleme sürecini yönetmek için, muhtemelen ilgili kaynak kodu ve verilerle birlikte sunulan araştırma metinlerini hazırlayan herhangi bir kuruluş. Yazılım ve veriler, derginin politikasına bağlı olarak ana yayının bir parçası veya ek materyal olarak verilen varlıklar olabilir. Ek olarak, yayıncılar, konuya özel uzmanlar topluluğu tarafından yürütülen, kabul edilen araştırma materyalinin kalitesini sağlamak için (genellikle akran değerlendirmesi) bir süreç uygular.*
Araştırma Yazılımı	Herhangi bir disiplindeki araştırmacıların , bir araç, sonuç veya çalışma nesnesi olarak kabul *edilsin, bilimsel altyapı*

Bilimsel Depo	Makaleler, veri kümeleri, yazılım gibi araştırma eserlerini arşivlemek ve kullanıma sunmak için çağrılan bir kuruluş . Örnekler arasında HAL, Zenodo, figshare, Dryad bulunur
MELİ	RFC 2129'da olduğu gibi kullanılır: Bu kelime veya "TAVSİYE EDİLEN" sıfatı, belirli durumlarda belirli bir öğeyi göz ardı etmek için geçerli nedenlerin bulunabileceği, ancak farklı bir yol seçmeden önce tüm sonuçların anlaşılması ve dikkatlice tartılması gerektiği anlamına gelir.
SIRS	Araştırma Yazılımları için Bilimsel Altyapılar
Yazılım Kaynak Kodu	Genel Kamu Lisansı'nda çok kısaca belirtildiği gibi, “ Bir eserin kaynak kodu, üzerinde değişiklik yapmak için eserin tercih edilen şekli anlamına gelir.” Bu tanım, genellikle düz metin olarak yazılan, insan tarafından okunabilen talimatların yaygın durumunu içerir.
SWHID	Yazılım Mirası kalıcı Tanımlayıcı
Evrensel Yazılım Arşivi	Nerede, neden ve nasıl geliştirildiğinden bağımsız olarak, kamuya açık tüm yazılım kaynak kodlarını toplayan, koruyan ve bunlara erişim sağlayan bir arşiv tutan bir kuruluş. Şu anda, Yazılım Mirası Vakfı'nın rolü budur.
VCS	Sürüm kontrol sistemi: geliştiriciler tarafından zaman içinde kaynak kodunda yapılan değişiklikleri izlemek ve yönetmek için kullanılan yazılım araç seti.

6.2. bibliyografya

Abelson, H., & Sussman, GJS, J. (1985). *Bilgisayar Programlarının Yapısı ve Yorumlanması* . MİT Basın.

Abramatic, J.-F., Di Cosmo, R. ve Zacchiroli, S. (2018a). Kaynak kodun evrensel arşivini oluşturmak. *ACM Haberleşmeleri* , *61* (10), 29–31. https://doi.org/10.1145/3183558

Abramatic, J.-F., Di Cosmo, R. ve Zacchiroli, S. (2018b). Kaynak Kodunun Evrensel Arşivini Oluşturma. *Komün. ACM* , *61* (10), 29–31. https://doi.org/10.1145/3183558

ACM. (SD.). *Veri ve yazılım artefaktları* . https://www.acm.org/publications/artifacts

ACM. (2016). *Artefakt incelemesi ve rozetleme* . https://www.acm.org/publications/policies/artifact-review-and-badging

Allen, A., Aragon, CR, Becker, C., Carver, J., Chis, A., Combemale, B., Croucher, M., Crowston, K., Garijo, D., Gehani, A., Goble , CA, Haines, R., Hirschfeld, R., Howison, J., Huff, KD, Jay, C., Katz, DS, Kirchner, C., Kuksenok, K., … Vinju, JJ (2017). Mühendislik Akademik Yazılımı (Dagstuhl Perspectives Workshop 16252). *Dagstuhl Manifestosu* , *6* (1), 1–20. https://doi.org/10.4230/DagMan.6.1.1

Alliez, P., Di Cosmo, R., Guedj, B., Girault, A., Hacid, M.-S., Legrand, A., & Rougier, N. (2020a). İlişkilendirme ve Referans Verme (Araştırma) Yazılımı: Inria'dan En İyi Uygulamalar ve Outlook. *Bilim ve Mühendislikte Bilgi İşlem* , *22* (1), 39-52. https://doi.org/0.1109/MCSE.2019.2949413

Alliez, P., Di Cosmo, R., Guedj, B., Girault, A., Hacid, M.-S., Legrand, A., & Rougier, N. (2020b). İlişkilendirme ve Referans Verme (Araştırma) Yazılımı: Inria'dan En İyi Uygulamalar ve Outlook. *Fen Mühendisliğinde Hesaplama* , *22* (1), 39-52. https://doi.org/0.1109/MCSE.2019.2949413

Arévalo, M., Escobar, C., Monasse, P., Monzón, N., & Colom, M. (2017). IPOL Demo Sistemi: Tekrarlanabilir Araştırma için Ölçeklenebilir Mikro Hizmetler Mimarisi. B. Kerautret, M. Colom ve P. Monasse (Eds.), *Reproducible Research in Pattern Recognition* (Cilt 10214, s. 3-16) içinde. Springer Uluslararası Yayıncılık. https://doi.org/10.1007/978-3-319-56414-2_1

Baker, M. (2016a). 1.500 bilim insanı, tekrarlanabilirlik konusundaki kapağı kaldırıyor. *Doğa* , *533* (7604), 452-454. https://doi.org/10.1038/533452a

Baker, M. (2016b). 1.500 bilim insanı, tekrarlanabilirlik konusundaki kapağı kaldırıyor. *Doğa* , *533* (7604), 452-454. https://doi.org/10.1038/533452a

Barborini, Y., Cosmo, RD, Dumont, AR, Gruenpeter, M., Marmol, BP, Monteil, A., Sadowska, J., & Zacchiroli, S. (2018). *Yeni bir bilimsel birikim türünün yaratılması: Yazılım* . https://www.softwareheritage.org/wp-content/uploads/2020/01/barborini-rda-poster.pdf https://hal.inria.fr/hal-01738741

Barnes, N. (2010). *Bilim kodu manifestosu* . http://sciencecodemanifesto.org

Benureau, FCY ve Rougier, NP (2018). Yeniden Çalıştır, Tekrarla, Yeniden Üret, Yeniden Kullan, Çoğalt: Kodu Bilimsel Katkılara Dönüştürmek. *Nöroinformatikte Sınırlar* , *11* . https://doi.org/10.3389/fninf.2017.00069

Bilder, G., Lin, J. ve Neylon, C. (2015). *Açık bilimsel altyapılar için ilkeler-v1* . https://doi.org/10.6084/M9.FIGSHARE.1314859

Bönisch, S., Brickenstein, M., Greuel, G.-M., & Sperber, W. (2012a). SwMATH – matematiksel yazılımınız için alıntılar. *dergiId:00006143* , *2012* . https://doi.org/10.1007/BF03345852

Bönisch, S., Brickenstein, M., Greuel, G.-M., & Sperber, W. (2012b). SwMATH – matematiksel yazılımınız için alıntılar. *dergiId:00006143* , *2012* . https://doi.org/10.1007/BF03345852

Borgman, CL, Wallis, JC ve Mayernik, MS (2012). Veriler Kimde? Bilim ve Teknoloji İşbirliklerinde Karşılıklı Bağımlılıklar. *Bilgisayar Destekli Ortak Çalışma* , *21* (6), 485–523. https://doi.org/10.1007/s10606-012-9169-z

Bradner, SO (1997). *Gereksinim düzeylerini belirtmek için RFC'lerde kullanılacak anahtar sözcükler . 2119* . https://doi.org/0.17487/RFC2119

Buckheit, JB ve Donoho, DL (1995). WaveLab ve Tekrarlanabilir Araştırma. A. Antoniadis & G. Oppenheim (Eds.), Wavelets *and Statistics* (Cilt 103, s. 55-81). Springer New York. https://doi.org/10.1007/978-1-4612-2544-7_5

Burton, A., Fenner, M., Haak, W., & Manghi, P. (2017). *Bilimsel iletişim bağlantılarının değişimi için Scholix meta veri şeması* . https://doi.org/10.5281/zenodo.1120265

CASRAI. (2015). *CREDIT Taksonomisi* . https://casrai.org/credit/

Chawla, DS (2016). Bilimsel yazılımın gizli kahramanları. *Doğa* , *529* (7584), 115-116. https://doi.org/10.1038/529115a

Clément-Fontaine, M., Di Cosmo, R., Guerry, B., moreau, P., & Pellegrini, F. (2019). *Araştırmadan elde edilen yazılımın daha geniş kullanımını teşvik etmek* [Araştırma raporu]. Açık Bilimin Özgür Yazılım ve Açık Kaynak Proje Grubu Komitesi. https://hal.archives-ouvertes.fr/hal-02545142

COAR ve SPARC. (2019, Ocak). *Bilimsel iletişim hizmetleri için iyi uygulama ilkeleri* . https://www.coar-repositories.org/news-updates/good-practice-principles-for-scholarly-communication-services-2/

Collberg, CS ve Proebsting, TA (2016). Bilgisayar sistemleri araştırmalarında tekrarlanabilirlik. *Komün. ACM* , *59* (3), 62-69. https://doi.org/10.1145/2812803

Colom, M., Kerautret, B. ve Krähenbühl, A. (2019). Tekrarlanabilir Araştırma ve Artırılmış Yayınlar için Platformlara Genel Bir Bakış. Bertrand Kerautret, M. Colom, D. Lopresti, P. Monasse ve H. Talbot (Eds.), *Reproducible Research in Pattern Recognition* (Cilt 11455, s. 25-39) içinde. Springer Uluslararası Yayıncılık. https://doi.org/10.1007/978-3-030-23987-9_2

Colom, M., Kerautret, B., Limare, N., Monasse, P., & Morel, J.-M. (2015). IPOL : Tamamen tekrarlanabilir araştırmalar için yeni bir dergi; dört yıllık gelişimin analizi. *2015 7. Uluslararası Yeni Teknolojiler, Mobilite ve Güvenlik Konferansı (NTMS)* , 1-5. https://doi.org/10.1109/NTMS.2015.7266500

Dacos, M., Di Cosmo, R. ve Pellegrini, F. (2018). *Open Science Monitor'deki yazılım göstergeleri önerisi hakkında* . Özgür ve Açık Kaynak yazılım çalışma grubu, Ulusal Açık Bilim Komitesi. https://www.ouvrirlascience.fr/wp-content/uploads/2018/11/2018.09.19-About-the-proposal-for-software-indicators-in-OSM.pdf

DataCite Meta Veri Çalışma Grubu. (2017). *Araştırma Verilerinin Yayınlanması ve Alıntılanması için DataCite Meta Veri Şeması Belgeleri v4.1* [Uygulama/pdf]. 72 sayfa. https://doi.org/10.5438/0014

Di Cosmo, R. (2020a, mai). *Biblatex-Yazılım* . CTAN: /tex-archive/makros/lateks/katkı/biblatex-katkı/biblatex-yazılım. https://www.ctan.org/tex-archive/macros/latex/contrib/biblatex-contrib/biblatex-software

Di Cosmo, R. (2020b). Software Heritage ile kaynak kodunu arşivleme ve referans verme. *ICMS* , *12097* , 362-373. https://doi.org/10.1007/978-3-030-52200-1_36

Di Cosmo, R. ve Danelutto, M. (2020). [Rp] OCamlP3l deneyinin çoğaltılması ve kopyalanması. ReScience *C'de* (Cilt 6, Numero 1, s. #2). https://doi.org/10.5281/zenodo.3936402

Di Cosmo, R., Gruenpeter, M., Marmol, B., Monteil, A., Romary, L., & Sadowska, J. (2020). Araştırma yazılımı eserlerinin küratörlü arşivlenmesi : Fransız açık arşivinden (HAL) öğrenilen dersler. *International Journal of Digital Curation* , *15* (1), 16. https://doi.org/10.2218/ijdc.v15i1.698

Di Cosmo, R., Gruenpeter, M., Marmol, BP, Monteil, A., Romary, L., & Sadowska, J. (2019). *Araştırma Yazılımı Eserlerinin Küratörlü Arşivlenmesi : Fransız açık arşivinden (HAL) öğrenilen dersler* . https://hal.inria.fr/hal-02475835

Di Cosmo, R., Gruenpeter, M. ve Zacchiroli, S. (2020a). Kaynak kodu eserlerine referans verme : Yazılım alıntısında ayrı bir endişe. *Bilim ve Mühendislikte Bilgi İşlem* , *22* (2), 33-43. https://doi.org/10.1109/MCSE.2019.2963148

Di Cosmo, R., Gruenpeter, M. ve Zacchiroli, S. (2020b). Kaynak Kodu Yapılarına Referans Verme : Yazılım Alıntılarında Ayrı Bir Endişe. *Bilim ve Mühendislikte Bilgi İşlem* , *22* (2), 33-43. https://doi.org/10.1109/MCSE.2019.2963148

Di Cosmo, R., Gruenpeter, M. ve Zacchiroli, S. (2018). *Dijital Nesneler için Tanımlayıcılar: Yazılım Kaynak Kodu Koruma Örneği* . 1-9. https://doi.org/10.17605/OSF.IO/KDE56

ouvertes.fr/hal-01590958

Eghbal, N. (2016). *Yollar ve Köprüler : Dijital Altyapımızın Arkasındaki Görünmeyen Emek.* https://www.fordfoundation.org/work/learning/research-reports/roads-and-bridges-the-unseen-labor-behind-our-digital-infrastructure/

Fenner, M. (2020). *DataCite Commons—PID'lerin Gücünden ve PID Grafiğinden Yararlanma* . https://doi.org/10.5438/F4DF-4817

Fenner, M., Katz, DS, Nielsen, LH ve Smith, A. (2018). *Yazılım için DOI kayıtları* . https://doi.org/10.5438/1NMY-9902

Açık Bilim için Fransız Ulusal Komitesi. (2019). *Platformlar, altyapılar ve editoryal içerik aracılığıyla Ulusal Açık Bilim Fonu'ndan fon sağlamak için Örneklik Kriterleri.* https://www.ouvrirlascience.fr/examplarity-criteria-for-funding-from-the-national-open-science-fund/

Gil, Y., H. David, C., Demir, I., Essawy, B., Fulweiler, W., Goodall, J., Karlstrom, L., Lee, H., Mills, H., Oh, J .-H., Pierce, S., Pope, A., Tzeng, M., Villamizar, S., & Yu, X. (2016). Geleceğin Yerbilimi Belgesine Doğru : Veriden Yazılıma ve Kökene Kadar Araştırmaları Belgelemek ve Paylaşmak İçin En İyi Uygulamalar : Geleceğin Yerbilimi Belgesi. *Yer ve Uzay Bilimi* , *3* . https://doi.org/10.1002/2015EA000136

Gruenpeter, M. ve Sadowska, J. (2018). *Bir Yazılım Yatırmasının Denetlenmesi* [Teknik rapor]. Hindistan ; CCSD ; Yazılım Mirası. https://hal.inria.fr/hal-01876705

Hinsen, K. (2019). Yazılım çöküşü ile başa çıkmak. *Fen Mühendisliğinde Hesaplama* , *21* (3), 104-108. https://doi.org/0.1109/MCSE.2019.2900945

Hinsen, K. (2013). Tekrarlanabilir Araştırma için Yazılım Geliştirme. *Bilim ve Mühendislikte Hesaplama* , *15* (4), 60-63. https://doi.org/10.1109/MCSE.2013.91

Howison, J. ve Bullard, J. (2015). Bilimsel literatürde yazılım : Biyoloji literatüründe bahsedilen yazılımları görme, bulma ve kullanma ile ilgili sorunlar. *Bilgi Bilimi ve Teknolojisi Derneği Dergisi* , *67* (9), 2137-2155. https://doi.org/10.1002/asi.23538

Avcı, JD (2007). Matplotlib : Bir 2D grafik ortamı. *Bilim ve Mühendislikte Hesaplama* , *9* (3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Jones, MB, Boettiger, C., Mayes, AC, Arfon Smith, Slaughter, P., Niemeyer, K., Gil, Y., Fenner, M., Nowak, K., Hahnel, M., Coy, L. , Allen, A., Crosas, M., Sands, A., Hong, NC, Cruse, P., Katz, D., & Goble, C. (2016). *CodeMeta : Yazılım meta verileri için bir değişim şeması. KNB Veri Deposu* [Application/ld+json]. KNB Veri Deposu. https://doi.org/10.5063/SCHEMA/CODEMETA-1.0

Katz, DS ve Clark, T. (2019). *Veri alıntılama ve yazılım alıntılama uygulamalarının karşılaştırılması ve analizi* . https://doi.org/10.5281/ZENODO.3497122

Katz, DS, Hong, NPC, Clark, T., Muench, A., Stall, S., Bouquin, D., Cannon, M., Edmunds, S., Faez, T., Feeney, P., Fenner, M., Friedman, M., Grenier, G., Harrison, M., Heber, J., Leary, A., MacCallum, C., Murray, H., Pastrana, E., … Yeston, J. (2020). Yazılım atıfının önemi. *F1000Research* , *9* , 1257. https://doi.org/10.12688/f1000research.26932.1

Knoth, P. (s. d.). *ÇEKİRDEK* . https://core.ac.uk/

Krishnamurthi, S. (2011). *Yazılım konferansları için artifakt değerlendirmesi* . https://www.artifact-eval.org/

Lamprecht, A.-L., Garcia, L., Kuzak, M., Martinez, C., Arcila, R., Martin Del Pico, E., Dominguez Del Angel, V., van de Sandt, S., Ison , J., Martinez, PA, McQuilton, P., Valencia, A., Harrow, J., Psomopoulos, F., Gelpi, J. Ll., Chue Hong, N., Goble, C., & Capella-Gutierrez , S. (2020). Araştırma yazılımı için FAIR ilkelerine doğru. *Veri Bilimi* , *3* (1), 37-59. https://doi.org/10.3233/DS-190026

Liang, W. ve Kai Yong , Z. (2013). Gen dizisini, istatistiksel makine çevirisine dayalı olarak gen ontolojisi terimlerine çevirin. *F1000Research* , *2* , 231. https://doi.org/10.12688/f1000research.2-231.v1

Liang, W. ve KaiYong , Z. (2013). *Gen Ontolojisi Çeviri Sistemine Protein Dizisi* . Zenodo. https://doi.org/10.5281/ZENODO.7506

Kodunuzu Alınabilir Hale Getirmek · GitHub Kılavuzları . (SD.). Consulté 7 Ekim 2020, à l'addresse https://guides.github.com/activities/citable-code/

McDougal, RA, Morse, TM, Carnevale, T., Marenco, L., Wang, R., Migliore, M., Miller, PL, Shepherd, GM, & Hines, ML (2016). Yirmi yıllık ModelDB ve ötesi: Sinirbilimin geleceği için temel modelleme araçları oluşturmak. *Hesaplamalı Sinirbilim Dergisi* , *42* (1), 1–10. https://doi.org/10.1007/s10827-016-0623-7

Toplantı, EG (2019). *Paris Çağrısı: Sürdürülebilir Kalkınma Mirası Olarak Yazılım Kaynak Kodu* . https://unesdoc.unesco.org/ark:/48223/pf0000366715

Moher, D., Bouter, L., Kleinert, S., Glasziou, P., Sham, MH, Barbour, V., Coriat, A.-M., Foeger, N., & Dirnagl, U. (2020) . Araştırmacıları değerlendirmek için hong kong ilkeleri: Araştırma bütünlüğünü teşvik etmek. *PLOS Biyoloji* , *18* (7), e3000737. https://doi.org/10.1371/journal.pbio.3000737

Muench, A., Accomazzi, A. ve Holm Nielsen, L. (2017). *Asklepias: Yazılım Alıntılama ve Keşif İş Akışlarını Etkinleştirme* . https://doi.org/10.5281/ZENODO.803473

Pan, X., Yan, E., Cui, M., & Hua, W. (2019). Yazılım, kütüphane ve bilgi bilimi araştırmaları için ne kadar önemlidir? Tam metin yayınların içerik analizi. *Journal of Informetrics* , *13* (1), 397-406. https://doi.org/10.1016/j.joi.2019.02.002

Paskin, N. (2010). Dijital nesne tanımlayıcı (DOI) sistemi. *Kütüphane ve bilgi bilimleri ansiklopedisi* , *3* , 1586–1592. https://www.taylorfrancis.com/books/e/9780203757635/bölüm/10.1081/E-ELIS3-12044418

Peroni, S. ve Shotton, D. (2020). OpenCitations, açık burs için bir altyapı organizasyonu. *Nicel Bilim Çalışmaları* , *1* (1), 428-444. https://doi.org/10.1162/qss_a_00023

Tüy, T. (2013). *Plum projesi* . https://projet-plume.org/types-de-fiches#logiciel_valide

Press, W. (1986). *Sayısal tarifler : Bilimsel hesaplama sanatı* . Cambridge Üniversitesi Yayınları.

Araştırma Verileri Alliance/FORCE11 Yazılım Kaynak Kodu Tanımlama WG, Allen, A., Bandrowski, A., Chan, P., Di Cosmo, R., Fenner, M., Garcia, L., Gruenpeter, M., Jones, CM , Katz, DS, Kunze, J., Schubotz, M., & Todorov, IT (2020). *Kalıcı yazılım kaynak kodu tanımlaması (V1.0) için vakaları ve tanımlayıcı şemaları kullanın* . https://doi.org/10.15497/RDA00053

Roberts, KV (1969). Bilimsel fortran programlarının yayınlanması. *Bilgisayar Fiziği İletişimi* , *1* (1), 1–9. https://doi.org/10.1016/0010-4655(69)90011-3

Rougier, NP, Hinsen, K., Alexandre, F., Arildsen, T., Barba, LA, Benureau, FCY, Brown, CT, de Buyl, P., Çağlayan, O., Davison, AP, Delsuc, M. -A., Detorakis, G., Diem, AK, Drix, D., Enel, P., Girard, B., Guest, O., Hall, MG, Henriques, RN, … Zito, T. (2017). Sürdürülebilir hesaplama bilimi: ReScience girişimi. *PeerJ Bilgisayar Bilimi* , *3* , e142. https://doi.org/10.7717/peerj-cs.142

Rousseau, G., Di Cosmo, R. ve Stefano Zacchiroli. (2020). Genel kaynak kodu ölçeğinde yazılım kaynağı izleme. *Ampirik Yazılım Mühendisliği* , 1-30. https://doi.org/10.1007/s10664-020-09828-5

Araştırma Değerlendirmesi Üzerine San Francisco Deklarasyonu (DORA) . (2013). https://sfdora.org/

Seinstra, F., Wallom, D. ve Keahey, K. (2015). Editoryal. *SoftwareX* , *1-2* , 1–2. https://doi.org/10.1016/j.softx.2015.08.001

Shustek, LJ (2006). Yazılım Tarihini Korumak İçin Neleri Toplamalıyız? *IEEE Annals of the History of Computing* , *28* (4), 110–112. https://doi.org/10.1109/MAHC.2006.78

Smith, AM, Katz, DS ve Niemeyer, KE (2016). Yazılım atıf ilkeleri. *PeerJ Bilgisayar Bilimi* , *2* , e86. https://doi.org/10.7717/peerj-cs.86

Smith, AM, Niemeyer, KE, Katz, DS, Barba, LA, Githinji, G., Gymrek, M., Huff, KD, Madan, CR, Mayes, AC, Moerman, KM, Prins, P., Ram, K ., Rokem, A., Teal, TK, Guimera, RV ve Vanderplas, JT (2018). Açık Kaynak Yazılım Dergisi (JOSS): Tasarım ve ilk yıl incelemesi. *PeerJ Bilgisayar Bilimi* , *4* , e147. https://doi.org/10.7717/peerj-cs.147

Spinellis, D. (2003). Web referanslarının bozulması ve başarısızlıkları. *ACM Haberleşmeleri* , *46* (1), 71–77. https://doi.org/10.1145/602421.602422

Stodden, V., LeVeque, RJ ve Mitchell, I. (2012). Bilimsel Hesaplama için Tekrarlanabilir Araştırma: Kültürü Değiştirmek İçin Araçlar ve Stratejiler. *Bilim ve Mühendislikte Hesaplama* , *14* (4), 13–17. https://doi.org/0.1109/MCSE.2012.38

Stojnic, R., Taylor, R., Karadaş, M., Kerkez, V., & Viaud, L. (2019). *Kodlu Kağıtlar* . https://paperswithcode.com/about

Sun, S., Lannom, L. ve Boesch, B. (2003). *Kol sistemine genel bakış* (RFC No 3650).

UNESCO, Genel Müdürlük (2020). Açık Bilim Tavsiyesinin (CL/4333) *ilk taslağı hakkında ön rapor* . UNESCO. https://unesdoc.unesco.org/ark:/48223/pf0000374409

UNESCO, E. grup toplantısı. (2019). *Paris Çağrısı: Sürdürülebilir Kalkınma Mirası Olarak Yazılım Kaynak Kodu* . https://unesdoc.unesco.org/ark:/48223/pf0000366715

Van Noorden, R., Maher, B. ve Nuzzo, R. (2014). En iyi 100 gazete. *Doğa* , 550-553. https://doi.org/10.1038/514550a

Wagner, M. (2017). *Dartla Boğa Gözüne Vurmak: Bilgisayar bilimlerinde artifakt değerlendirmesi* . https://doi.org/10.5281/ZENODO.583007

Wilkinson, MD, Dumontier, M., Aalbersberg, Ij. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, LB, Bourne, PE ve diğerleri. (2016). Bilimsel veri yönetimi ve yönetimi için ADİL Yol Gösterici İlkeler. *Bilimsel veriler* , *3* . https://doi.org/10.1038/sdata.2016.18

Zhao, R. ve Wei, M. (2017). Açık kaynaklı yazılımların etki değerlendirmesi: Bir Altmetrics perspektifi. *Scientometrics* , *110* (2), 1017-1033. https://doi.org/10.1007/s11192-016-2204-y

6.3. Görev Gücü katılımcıları

6.3.1. Roberto Di Cosmo (Başkan TF SIRS)

Organizasyon(lar)
Software Heritage, Inria tarafından Unesco ile ortaklaşa başlatılan ve büyük BT oyuncuları, devlet kurumları ve akademik kuruluşlar dahil olmak üzere çeşitli paydaşlar tarafından desteklenen, kar amacı gütmeyen açık bir girişimdir. Belirtilen amacı, evrensel bir kaynak kodu yazılım bilgi tabanı oluşturarak, tam geliştirme geçmişi ile şimdiye kadar yazılmış tüm yazılımların kaynak kodunu toplamak, korumak ve paylaşmaktır. Software Heritage, çeşitli ihtiyaçlara hitap eder: bilimsel ve teknolojik bilgimizi korumak, daha iyi yazılım geliştirmeyi ve toplum ve endüstri için yeniden kullanılmasını sağlamak, daha iyi bilimi teşvik etmek ve büyük ölçekli, tekrarlanabilir yazılım çalışmaları için gerekli bir altyapıyı oluşturmak. 140 milyondan fazla depodan 9 milyardan fazla benzersiz kaynak dosyasıyla, şimdiye kadar oluşturulmuş en büyük kaynak kodu arşividir. Daha fazla bilgi: https://www.softwareheritage.org .

Kısa Özgeçmiş
Scuola Normale Superiore di Pisa mezunu olan Roberto Di Cosmo, Paris'teki Ecole Normale Supérieure'de doçent, daha sonra Paris Üniversitesi'nde Bilgisayar Bilimleri profesörüydü ve şu anda Inria'da izinli, 20'den fazla uluslararası dergi makalesi ve 50 uluslararası yayın yapıyor. teorik hesaplama, fonksiyonel ve paralel programlama ve yazılım mühendisliğinde konferans makaleleri.

Systematic'in Özgür Yazılım tematik grubunu ve Açık Kaynak Yazılımına adanmış bir araştırma yapısı olan IRILL'i oluşturduktan sonra, ortaklaşa, halka açık tüm kaynak kodların evrensel arşivini oluşturma misyonuyla Yazılım Mirasını oluşturmak için Inria'dan destek aldı. UNESCO ile.

Yazılım Mirasında Rol: Kurucu ve Yönetici

Daha fazla bilgi: https://www.dicosmo.org/bio.html

SIRS TF'ye ilgi
Eski bir açık erişim ve açık kaynak savunucusu olarak Roberto, araştırmanın açık, şeffaf ve tekrarlanabilir olması gerektiğine kuvvetle inanmaktadır. Bu vizyonu destekleyecek ortak altyapıların eksikliğini hissetmiştir ve özellikle yazılımın araştırma ve Açık Bilimin temel direği olarak çok uzun süredir göz ardı edildiğini hissetmiştir, ancak bunun olumlu bir yanı da vardır: Hâlâ işin içinde olabiliriz. (araştırma) yazılımı için bilimsel altyapılar için yalın, verimli, açık, paylaşılan, ortaklaşa, işbirlikçi bir mimariyi savunma ve on yıllardır diğer alanlarda görülen çabaların balkanlaşmasından ve dağılmasından kaçınma zamanı. Roberto, SIRS TF'nin bu hedefe katkıda bulunmak için büyük bir fırsat olduğuna inanıyor.

6.3.2. Jose Benito Gonzalez Lopez (TF SIRS Eşbaşkanı)

yayınlar, vb.) paylaşmalarını, korumalarını ve sergilemelerini sağlayan genel amaçlı bir havuzdur. Güvenilir CERN veri merkezinde kurulmuştur ve finansman ayrıca OpenAIRE (ana ortak) aracılığıyla EC, SLOAN vakfı ve Arcadia gibi diğer kaynaklardan gelse de, CERN tarafından yönetilir, geliştirilir ve sürdürülür. Zenodo, toplamda 1,5 milyondan fazla kayıt, yaklaşık 100.000 yazılım kaydı (tüm sürümler dahil) ve yaklaşık 350 TB dosya barındırıyor.

Kısa
özgeçmiş Jose Benito Gonzalez Lopez, jose.benito.gonzalez@cern.ch

Şu anda Jose , CERN'de, Bilimsel Depolar ve Açık Bilim Altyapısı ile ilgili geliştirme (Açık Kaynak Yazılım) ve hizmet sağlamadan sorumlu çeşitli ekiplerden oluşan Dijital Depolar Bölümünü yönetmektedir. Bunlar ana projeler:

- CERN Doküman Sunucusu (CDS), CERN kurumsal depoları:
  - https://cds.cern.ch
  - https://videos.cern.ch
- Aşağıdakilerden sorumlu olan Dijital Bellek (DM) projesi:
  - dijitalleşme
  - Uzun süreli koruma
  - Bakınız: http://digital-memory-project.web.cern.ch
- Invenio Yazılımı:
  - http://inveniosoftware.org
- Aşağıdakilerden sorumlu olan Açık ve Tekrarlanabilir Araştırma (ORR):
  - https://opendata.cern.ch
  - https://reana.io
- Aşağıdakilerden sorumlu olan Zenodo:
  - https://zenodo.org

Jose, açık kaynaklı yazılım geliştirme ve proje yönetimi konusunda çok fazla deneyime sahip, eğitimli bir Yazılım Mühendisidir. Daha önce mevcut pozisyonundan önce Jose, Açık Kaynak projesi Indico < getindico.io > 'nun geliştirilmesine ve daha sonra yönetimine katkıda bulundu .

SIRS TF'ye ilgi
Jose, biyografisinden de anlaşılacağı üzere Açık Bilim ve Açık Kaynak Yazılımın güçlü bir savunucusudur. Avrupa'nın, aidiyetleri ne olursa olsun araştırmaları daha verimli, adil ve herkes için erişilebilir kılacak gerçek bir Açık Bilim fabrikasına doğru ilerlediğini görmek istiyor. Böyle bir hayali gerçekleştirmek için birkaç alanda başarılı olmak gerekiyor

6.3.3. Kay Graf

Organizasyon(lar)
ESCAPE (Avrupa Bilim Kümesi Astronomi ve Parçacık fiziği ESFRI araştırma altyapıları) bir EOSC küme projesidir. Misyonu, ilgili ESFRI projeleri ile Avrupa Açık Bilimi arasında işlevsel bir bağlantı kurmak için astronomi ve hızlandırıcı tabanlı parçacık fiziği alanındaki yeni nesil Araştırma Altyapıları Avrupa Strateji Forumu (ESFRI) tesislerinden oluşan tek bir işbirlikçi küme oluşturmaktır. Bulut (EOSC).

Kısa
özgeçmiş Kay Graf, Ph.D in fizik, kay.graf@fau.de

Kay, Almanya'daki Erlangen Üniversitesi'ndeki (FAU) Erlangen Astropartikül Fiziği Merkezi'nde (ECAP) kıdemli araştırmacı ve oradaki genel müdürdür. Araştırmaları, yüksek enerji fiziği ve astrofizik topluluklarını kapsayan astroparçacık fiziği alanındadır. Ayrıca, bir ESFRI olan KM3NeT nötrino deneyinin bilgi işlem ve yazılım koordinatörü olarak yazılım geliştirme, koordinasyon ve bakımda uzun bir geçmişe sahiptir .

Kay, öncelikle EOSC küme projesi ESCAPE içindeki çalışmasıyla ilgili olarak EAWG'nin bir üyesidir - burada bir açık bilim yazılımı ve hizmet havuzu (OSSR) üzerindeki çalışma paketini koordine eder. OSSR, bilimsel yazılım ve hizmetleri bilim topluluğuyla paylaşmak ve açık bilimi etkinleştirmek için sürdürülebilir bir açık erişim deposu olacaktır. Veri işleme ve analiz için astro-parçacık fiziği ile ilgili bilimsel yazılım ve hizmetlerin yanı sıra test veri setlerini, kullanıcı destek belgelerini, öğreticileri, sunumları ve eğitim etkinliklerini barındıracak.

SIRS TF'ye ilgi
Çoğunlukla topluluğa özel olan yazılım ve verilerin, tüm bilim ürünlerinin temelini oluşturmak için doğal olarak el ele gittiği, eksiksiz yazılım yaşam döngülerinin ele alındığı, en iyi uygulamaların paylaşıldığı ve ortak geliştirmeler yoluyla çapraz gübrelemenin yapıldığı bir topluluktan geliyor. yazılım algoritmalarının ve yazılım platformlarının yeniden kullanımı onun ana ilgi alanıdır. Tüm bu konular SIRS görev gücünün bir parçasıdır - bu nedenle Kay, EOSC'nin temel direklerinden biri olarak yazılım stratejileri oluşturmak için birlikte çalışmaya hevesliydi.

6.3.4. Miguel Kolom

Organizasyon(lar)
Miguel Colom École Normale Supérieure de Cachan'da (şimdi ENS Paris-saclay) Prof. Jean-Michel Morel tarafından ERC gelişmiş hibesi alındıktan sonra 2009 yılında kurulan Image Processing On Line'ı (IPOL) temsil eder. IPOL, algoritmaların tam matematiksel açıklamalarına özel olarak odaklanarak sinyal (esas olarak görüntü ve video) işleme hakkında hakemli makaleler yayınlar. IPOL'ün yıllık yayın sayısı, yılda yaklaşık 25 makale ile mütevazıdır. SCOPUS, DOAJ ve diğerleri tarafından indekslenir. Resmi bir Etki Faktörü henüz elde edilmedi, ancak Thomson Reuters Gelişen Atıf İndeksinde (Etki Faktöründen önceki bir ön adım). IPOL'deki her yayın, yalnızca makalenin PDF'sini değil, her ikisi de ücretsiz lisans altında kaynak kodunu da içerir.

IPOL web sitesi: https://www.ipol.im

Kısa özgeçmiş
Miguel , Centre Borelli'de (ENS Paris-Saclay, Fransa) görüntü işleme alanında bir araştırmacıdır ve farklı konularda üç doktora adayını denetlemektedir: JPEG artefaktlarına dayalı görüntülerde sahteciliklerin tespiti, gürültü analizine dayalı sahtekarlıkların tespiti ve tasarımın tasarımı. düzensiz interferometrik örneklemeye dayalı yeni bir uydu konsepti. Daha önce, doğal ve hiperspektral görüntülerin gürültü kestirimi ve gürültü giderme konularında çalıştı. Akademik geçmişi uygulamalı matematik ve bilgisayar bilimleridir. http://mcolom.info

IPOL'de Miguel, bir bölüm editörü ve dağıtılmış bir mikro hizmet mimarisi olan mevcut demo sisteminin tasarımcısıdır. Onu geliştiren ve sürdüren mühendis ekibini yönetir.

SIRS
TF'ye ilgi IPOL'ün başlangıcından bu yana Miguel, sistemin kalitesine ve uzun vadeli dayanıklılığına büyük ilgi duymuştur. Bu, tamamen doğru kabul edilene kadar birkaç bölümün yeniden uygulanmasına yol açmıştır. Bu TF'ye

Ayrıca, bu görev gücündeki pek çok platformun paylaştığı ortak sorunların çözümleri hakkında bilgi sahibi olmak: nasıl verimli bir şekilde arşivlenir, tekrarlanabilirliğin nasıl sağlanacağı, arşivleme sırasında farklı türde ayrıntı düzeyinin nasıl yönetileceği vb. IPOL, IPOL'un inandığı çözümler buldu. iyi, ama Miguel aynı sorunlara farklı platformlar tarafından farklı çözümler bilmekle ilgileniyor.

Ve son olarak, tavsiyeler açısından başkalarına faydalı olabilecek iyi uygulamalar hakkında sonuçlara varmak için çalışma platformları hakkında bilgi toplamak için bu girişimin kesinlikle gerekli olduğunu düşünüyor. Bir tür "tasarım kalıpları", ancak platform düzeyinde ve belirli görevler için (algoritmaların yürütülmesi, arşivleme, referans verme, vb.) iyi uygulamalar olarak anlaşılır.

6.3.5. Paolo Manghi

Kuruluş(lar)
OpenAIRE, Avrupa ve ötesinde Açık Bilim uygulamalarının uygulanmasını ve benimsenmesini desteklemek için ağ hizmetleri ve teknik hizmetler sunan kar amacı gütmeyen bir tüzel kişiliktir. Sunduğumuz temel teknik hizmetlerden biri, dünya çapındaki tüm bilimsel iletişim kaynaklarını bir araya getiren açık, şeffaf, üst veri koleksiyonu olan OpenAIRE Araştırma Grafiğidir . Yaklaşık 12.000 kaynaktan (Crossref, DataCite, Unpaywall, MAG, ORCID, GRID/ROR, ön baskılar, OpenDOAR'dan kurumsal depolar vb.) projeler, kuruluşlar (ve bunları topladığımız veri kaynakları). Grafik _ 110 Pub, 7 Mi veri seti, 200K yazılım, 30 fon sağlayıcı, 3,5 milyon proje ve bu tür nesneler arasındaki yaklaşık 1Bi ilişkiyi sayar. OpenAIRE, Avrupa Açık Bilim Bulutunun temel direklerinden biridir .

Kısa özgeçmiş
Paolo Manghi, Consiglio Nazionale delle Ricerche (CNR), Pisa, İtalya'daki Istituto di Scienza e Tecnologie dell'Informazione (ISTI) bilgisayar bilimi alanında (PhD) Araştırmacıdır. İlgi alanları, günümüzde bilim ve bilimsel iletişim altyapıları için veri e-altyapıları olup, farklı disiplinler içinde ve arasında açık bilim yayıncılığını destekleyen teknolojilere, yani hesaplamalı tekrarlanabilirlik ve bilimin şeffaf değerlendirilmesine odaklanmaktadır. OpenAIRE altyapısının CTO'su olup, OpenAIRE-Connect, OpenAIRE-Advance, OpenAIRE2020 H2020 projelerinde koordinasyon ve araştırmalarda yer almaktadır. Paolo ayrıca SoBigDataPlus araştırma altyapısı projelerinde yer almaktadır. , PARTHENOS, AriadnePlus, RISIS2 ve Avrupa Açık Bilim Bulutu projelerinde EOSCpilot, eInfraCentral, EOSC Secretariat, EOSC-Enhance. Research Data Alliance WG'lerinin aktif bir üyesi, ResearchObject.org , GreyNet, RD-Switchboard girişiminin EC projeleri danışma kurullarının üyesi, Avrupa Komisyonu için Open Science Monitor WG, EOSC Architecture WG, GO FAIR GO Inter WG, ve Dünya Veri Sistemi ITO Teknik Danışma Komitesi.

SIRS TF Paolo'ya
ilgi Temel araştırma ilgi alanları, tüm bilimsel sürecin paylaşılmasını, izlenmesini, izlenmesini, çoğaltılmasını, değerlendirilmesini ve ödüllendirilmesini sağlamak için Açık Bilim yayın iş akışlarına yönelik çözümler üzerinedir. Bu alandaki yakın tarih, Açık Erişim'den yayınlara, verilere Açık Erişim'e, FAIR Data'ya geçişe ve şimdi ilk kez bilimsel iletişimin birinci sınıf vatandaşı olarak yazılıma göz atmaya kadar bu sorunları ele alıyor. Bu adımın, bugün uygulanmaktan çok uzak, kapsayıcı bir bilim görüşüne doğru ilerlemek için gerekli ve anahtar olduğuna inanıyor. Açık Bilim yol haritasına bakıldığında, hizmetler, iş akışları, tesisler bilimsel iletişimin göz ardı ettiği kadar önemli olduğu için, yazılım bulmacanın tek eksik parçası olarak tasarlanmamalı, aksine şimdiye kadar göz ardı edilen "odadaki fil". .

6.3.6. melisa harrison

Organizasyon(lar)
eLife, fon sağlayıcılar tarafından oluşturulan ve araştırmacılar tarafından yönetilen kar amacı gütmeyen bir kuruluştur. Misyonu, en sorumlu davranışları teşvik eden ve tanıyan bir araştırma iletişimi platformu işleterek keşfi hızlandırmaktır.

eLife üç ana alanda çalışır:

- Yayıncılık – eLife, araştırmaların nasıl değerlendirildiğini ve yayınlandığını geliştirmek için yaratıcı yeni yollar keşfederken, biyoloji ve tıbbın tüm alanlarında en yüksek standartlara ve öneme sahip çalışmaları yayınlamayı amaçlar.

- Teknoloji – eLife, bilim yayıncılığı altyapısını modernize etmek ve yeni sonuçları paylaşmak, kullanmak ve bunlarla etkileşim kurmak için çevrimiçi araçları geliştirmek için açık kaynaklı teknoloji inovasyonuna yatırım

- Araştırma kültürü – eLife, araştırmada sorumlu davranışları teşvik etmek için dünya çapındaki araştırma topluluğuyla birlikte çalışmaya kendini adamıştır.

eLife, Howard Hughes Tıp Enstitüsü, Knut ve Alice Wallenberg Vakfı, Max Planck Topluluğu ve Wellcome'dan finansal destek ve stratejik rehberlik almaktadır. eLife Sciences Publications Ltd, açık erişimli eLife dergisinin yayıncısıdır.

Kısa Özgeçmiş
Melissa Harrison , üretim sürecini yöneterek, kabulden yayına ve sonraki teslimatlara kadar içeriği yöneten, verimli ve yenilikçi olan üretim departmanını yönetir. Çeşitli yayınevlerinde dergilerde ve kitaplarda editörlük ve üretim rolleri üstlenerek, dergi üretim aşamasında iş akışı, süreç ve XML tercihini pekiştirdi. Melissa, JATS4R'ye başkanlık eder ve Force11, Metadata2020, Crossref ve JATS topluluklarına düzenli olarak katkıda bulunur, bilginin yayılmasını ve yeniden kullanımını en üst düzeye çıkarmak için kullanıcılar arasında bilgi akışını kolaylaştırmak için veri modellemenin standartlaştırılması için kampanya yürütür. Kişileri, kurumları, kaynakları ve çıktıları birbirine bağlamak için PID'lerin kullanımı da dahil olmak üzere, açık bilim çıktılarının makine tarafından okunabilirliğini destekler.

SIRS TF'ye ilgi
eLife, en iyi uygulama, açık bilim ve tekrarlanabilirlik standartlarını uyguladığı ve diğer yayıncılarla olan çabalara öncülük ettiği için, yazılım "yazarlarına" kredi vermenin yanı sıra yazılımın kullanımı ve yeniden kullanılmasının uygulanması çok zor olmuştur ve gezin. Melissa, "doğru şeyi" yapmaya çalışan bir yayıncının bakış açısını ve bunun içerdiği sorunları ve diğer dergilerde alımı artırmak için neden bunu basitleştirmeye ihtiyaç duyulduğunu vermeyi umuyor.

6.3.7. Yannick Barborini

Kuruluş(lar)
HAL, Fransız üniversiteleri, en üst düzey üniversiteler ve araştırma kuruluşları tarafından bir kuruluşlar arası anlaşmanın (2013) parçası olarak seçilen ve araştırmacılarının bilimsel üretimlerini saklamalarına izin vermek için seçilen ulusal multidisipliner açık arşiv platformudur. HAL, Denetleme makamları CNRS, Inria, INRAE ve Lyon Üniversitesi olan bir ortak hizmet birimi (UMS 3668) olan Doğrudan Bilimsel İletişim Merkezi (CCSD) tarafından ve Yüksek Öğrenim, Araştırma ve Yenilik (MESRI). HAL, "2018-2020 Ulusal Araştırma Altyapıları Planı"nda yer alan altyapıların bir parçasıdır.

HAL, tüm bilimsel alanlarla ilgili araştırma yoluyla üretilen belgeleri (hakemli dergilerde yayınlanan makaleler, yayınlanmamış makaleler, iletişimler vb.) açık erişim yoluyla toplar ve dağıtır. HAL, bugüne kadar yaklaşık 730.000 bilimsel belge ve 2.300.000 bibliyografik kayıt içermektedir. Belge mevduatları (2019'da 94.246) yılda yaklaşık %20 büyüyor.

HAL platformu ayrıca 140 yüksek öğrenim ve araştırma kurumu portalına ev sahipliği yapıyor. Fransız üniversitelerinin, araştırma kuruluşlarının ve üst düzey üniversitelerin yarısından fazlası. Bu portallar, bu kuruluşların kurumsal açık arşivlerini oluşturmakta ve Açık Erişim politikalarını uygulamalarına ve üretimlerini yönetmelerine olanak sağlamaktadır.

Kısa özgeçmiş
Yannick Barborini bir yazılım mühendisidir ve 2005'ten beri CNRS'de çalışmaktadır. Tüm bilim camiasına sunulan farklı hizmetlerin geliştirilmesine katılmıştır: Elbette HAL, ayrıca Sciencesconf (konferans yönetim platformu), Episciences (overlay) dergiler) ve Isidore (Beşeri ve Sosyal Bilimlerde arama motoru). Şu anda, açık arşiv HAL için yeni hizmetler geliştirmekten ve sağlamaktan sorumlu ekibi yönetiyor.

SIRS'e ilgi TF
Yannick, araştırmanın açık ve erişilebilir olması gerektiğine inanıyor. Bilimsel çıktılar artık belgelerle (makaleler, iletişimler vb.) sınırlı değildir. HAL, CCSD, HAL Inria ve Software Heritage arasındaki işbirliği sayesinde 2018'de yazılım depolarını açtı, ancak arşivdeki yazılım öğelerinin yönetiminin iyileştirilmesi gerektiğini gördüler.

Yazılımlar, diğer bilimsel ürünler gibi birinci sınıf vatandaş olarak görülmelidir ve Yannick, bu görev gücünün diğer ortaklarla deneyimlerini paylaşmak ve bu amaca katkıda bulunmak için büyük bir fırsat olduğuna inanmaktadır.

Organizasyon(lar)
Ville Tenhunen, Finlandiya eğitim ve kültür bakanlığı tarafından aday gösterilen EOSC ÇG Mimarlık'ta Finlandiya üye devlet temsilcisidir. Finlandiya idari uygulamalarına göre bu, herhangi bir resmi kuruluşu temsil etmediği, bunun yerine kendisini, özellikle de kendi uzmanlığını temsil ettiği anlamına gelir. Ville, Helsinki Üniversitesi'nde çalıştı ve şimdi, bu yılın Mart ayının başından beri Amsterdam'daki EGI Vakfı'nda Veri Çözümleri Mimarı olarak çalışıyor. EGI Vakfı ile geçici bir sözleşmesi var.

Kısa Özgeçmiş
Ville, 12 yıldan fazla bir süredir Helsinki Üniversitesi'nde takım lideri ve proje yöneticisi olarak çalıştı. Son büyük proje , araştırma verileri ve depoları ile ilgilendi. O da Finlandiya ulusal açık bilim ve araştırma Girişimlerinde aktif olmuştur. Ayrıca Ville, Research Data Alliance'ın (RDA) Research Institutions IG'deki Research Data Architectures'ın eş başkanıdır ve şu anda EOSC'nin Mimari Çalışma Grubunun bir üyesidir. Mart 2020'den itibaren EGI Vakfı'nda Veri Çözümleri Mimarı olarak çalıştı. Ayrıca uluslararası bir araştırma projesi için (22 ülke ekibi) bazı çözümleri kodladığı APIKS projesinde veri yöneticisi olarak görev almıştır.

Helsinki Üniversitesi'nde ve şimdi EGI Vakfı'nda hizmet sağlayıcı ve hizmet geliştirici rollerinde bilimsel yazılımlarla çalıştı.

SIRS TF'ye ilgi
Ville, araştırma tekrarlanabilirliği ve açıklığı bağlamında yazılım koruma, keşfedilebilirlik hizmetleri ve PID sistemleriyle ilgilenmektedir. Katalog hizmetleri bu şekilde bir noktadır. Diğer ilgi alanları, örneğin kapsayıcılar, sanal cihazlar ve bir hizmet olarak işlevlerin tartışıldığı yeni hizmet ve yazılım biçimleridir.

6.3.9. Michael Wagner

Organizasyon(lar)
Schloss Dagstuhl - Leibniz-Zentrum für Informatik (İngilizce: Leibniz Enformatik Merkezi). Schloss Dagstuhl'un çok genel misyonu, bilişim alanında temel ve uygulamaya yönelik araştırmaları teşvik etmek, ileri düzeyde, bilimsel mesleki eğitimi ve bilişim alanında ileri eğitimi desteklemek, araştırma arasında bilişime bilgi aktarımını ve bunların uygulanması arasında bilgi aktarımını teşvik etmektir. bilişim ve bilişim için uluslararası bir forum ve araştırma enstitüsü işletmek.

Dagstuhl, misyonunu esas olarak araştırmacılar arasındaki iletişimi ve etkileşimi kolaylaştırarak sürdürmüştür. Dagstuhl'un 1990'daki ilk günlerinden beri, seminer ve çalıştay toplantı programı her zaman programatik çalışmalarının odak noktası olmuştur. Son yıllarda, LZI Schloss Dagstuhl operasyonunu genişletti ve ayrıca bibliyografik hizmetlerde (dblp bilgisayar bilimi bibliyografyası) ve açık erişim yayıncılığında önemli çabalar gösteriyor.

dblp: dblp bilgisayar bilimi bibliyografyası, başlıca bilgisayar bilimi dergileri ve tutanakları hakkında açık bibliyografik bilgiler sağlar. 5,1 milyondan fazla yayını listeleyen dblp, bilgisayar bilimi araştırma makalelerinin dünyanın en kapsamlı açık veri koleksiyonudur. https://dblp.org/ adresine bakın .

Yayıncılık: Dagstuhl, başlangıcından bu yana, ister kağıt üzerinde ister elektronik ortamda ücretsiz olarak sunulan seminer ve çalıştayların raporlarını yayınlamaktadır. Bilgisayar bilimlerinde açık erişim fikrinin yavaş başlamasına tepki olarak ve topluluktan tekrarlanan talepler aldıktan sonra, Dagstuhl 2008 yılında bilgisayar bilimi araştırmaları için bir açık erişim yayın platformu başlattı. Amaç, büyük bir yayınevi olmak değil, bilgisayar bilimlerinde uygun bir yayın biçimi olarak uygun fiyatlı açık erişim yayıncılığı kurmaktır.

Dagstuhl Publishing'in amiral gemisi ürünü, olağanüstü bilgisayar bilimi konferanslarının bildirilerini yayınlayan LIPics serisidir. Bakınız https://www.dagstuhl.de/dagpub

Kısa Özgeçmiş
Michael Wagner, 2012'den beri Schloss Dagstuhl'un bilimsel kadrosunun bir üyesidir. Orada dblp ekibinin bir parçası olarak başladı, ancak 2012'nin sonunda büyüyen yayıncılık departmanındaki ilk görevini hızla üstlendi. 2017'nin sonundan beri, Michael şimdi tam zamanlı yayıncılık departmanına liderlik ediyor. Operasyonel yayıncılık işine ek olarak, bir meslektaşı ile birlikte yazılım sistemlerinin geliştirilmesinden sorumludur.

SIRS TF'ye ilgi
Michael'ın sahip olduğu bilimin açık olması gerektiğine dair güçlü bir görüş. Tüm katkılar - metin, veri veya yazılım olsun - halka ücretsiz olarak sunulmalıdır. Ayrıca araştırma sonuçlarının, birincil amacı karı maksimize etmek olan ticari hizmet sağlayıcıların elinde olmaması gerektiğini düşünüyor. Yayıncılık işinde bu yavaş yavaş değişiyor gibi görünse de (en azından açık erişim kısmı yavaş yavaş artıyor; uygun fiyatlı yayın maliyeti ne yazık ki bir başka zor konu), şimdi

6.3.10. Wolfgang Dalitz

Organizasyon(lar)
Zuse Institute Berlin (ZIB): ZIB, uygulamalı matematik ve veri yoğun yüksek performanslı bilgi işlem için disiplinler arası bir araştırma enstitüsüdür. Araştırmaları, akademi ve endüstriden bilimsel işbirliği ortaklarıyla modelleme, simülasyon ve optimizasyon üzerine odaklanmaktadır.

Kısa Özgeçmiş
Wolfgang Dalitz, Zuse Institut Berlin'de (ZIB) bilimsel bilgi sistemleri alanında çalışan bir bilim insanıdır. 1980'lerin sonlarından beri matematiksel yazılım kütüphaneleri oluşturmaya dahil olmuştur. "Matematiksel Algoritmik Zeka" bölümünde, "Açık Bilim ve Araştırma Verileri" Çalışma Grubuna liderlik etmektedir.

SIRS TF'ye ilgi
Verilere, kodlara, yöntemlere ve bilimsel araştırma sonuçlarına açık erişim, ancak onları ilişkilendirmek ve birbirine bağlamak mümkün olduğunda tam potansiyelini ortaya çıkarır. Buna dayanarak, bilimsel ve sosyal önemi olan araçlar geliştirilebilir. Bunun için ön koşul, FAIR ilkelerine (Bulunabilir, Erişilebilir, Birlikte Çalışabilir, Yeniden Kullanılabilir) dayalı uygun bilimsel altyapıların varlığıdır. Bu TF'deki ortaklar bu ilkeleri desteklemektedir. Bu altyapıyı kurmak için ortak bir strateji, TF'nin hedefi olmalıdır.

6.3.11. Jason Maassen

Organizasyon(lar)
Hollanda e-Bilim Merkezi, araştırma yazılımlarının geliştirilmesi ve uygulanması için Hollanda ulusal merkezidir. NWO (Hollanda Araştırma Konseyi) ve SURF (Hollanda eğitim ve araştırmasında ICT organizasyonu) tarafından finanse edilen kar amacı gütmeyen bir organizasyondur. Ana hedefi, tüm disiplinlerde Hollanda akademik araştırmalarında dijital yöntemlerin ve araştırma yazılımlarının kullanımını artırmaktır. Bunu yapmak için, e-Bilim Merkezi araştırma projelerine hem finansman hem de uzmanlık (KGB'ler şeklinde) sağlar. Ayrıca, FAIR yazılım önerileri (fair-software.eu), araştırma yazılımı dizini (research-software.nl), yazılım kalite kılavuzları (the-turing-way.netlify.app) gibi araştırma yazılımını çevreleyen birçok konuya katkıda bulunur.), yazılım marangozluğu kursları vb.

Kısa Özgeçmiş
Jason Maassen bir Teknoloji Lideridir ve Hollanda eScience Center'daki bilimsel uygulamalara paralel ve dağıtılmış programlama uygulayan birçok projede yer almıştır. Ayrıca, Research Software Directory ve fair-software.nl gibi eScience Center'ın yazılım sürdürülebilirlik çabalarının merkezinde dahili yazılım geliştirmeye rehberlik eder.

SIRS TF'ye ilgi
Jason'ın ana ilgisi, Hollanda eScience Center'ın FAIR yazılımı, yazılım dizinleri, yazılım arşivleme, yazılım kalite yönergeleri vb. alanındaki çabalarını (potansiyel) Avrupalı ortakların ve EOSC'ninkilerle uyumlu hale getirmektir. Şimdiye kadar, e-Bilim Merkezi, burada ve orada geçici bazı uluslararası işbirliği ile çoğunlukla ulusal düzeyde aktif olmuştur. Jason, bu girişimi bu çabaların kapsamını genişletmek için bir fırsat olarak görüyor.

6.3.12. Carlos Martinez-Ortiz

Organizasyon(lar)
Hollanda e-Bilim Merkezi, araştırma yazılımlarının geliştirilmesi ve uygulanması için Hollanda ulusal merkezidir. NWO (Hollanda Araştırma Konseyi) ve SURF (Hollanda eğitim ve araştırmasında ICT organizasyonu) tarafından finanse edilen kar amacı gütmeyen bir organizasyondur. Ana hedefi, tüm disiplinlerde Hollanda akademik araştırmalarında dijital yöntemlerin ve araştırma yazılımlarının kullanımını artırmaktır. Bunu yapmak için, e-Bilim Merkezi araştırma projelerine hem finansman hem de uzmanlık (KGB'ler şeklinde) sağlar. Ayrıca, FAIR yazılım önerileri (fair-software.eu), araştırma yazılımı dizini (research-software.nl), yazılım kalite kılavuzları (the-turing-way.netlify.app) gibi araştırma yazılımını çevreleyen birçok konuya katkıda bulunur.), yazılım marangozluğu kursları vb.

Kısa Özgeçmiş
Carlos Martinez-Ortiz, dijital beşeri bilimler, yaşam bilimleri, binalardaki anormal enerji tüketiminin otomatik tespiti, süt ineklerinin video takibi, yüksek performanslı depolama sistemlerinin modellenmesi ve tıbbi görüntülerin segmentasyonuna kadar çok çeşitli araştırma projelerinde çalıştı. Topluluk yöneticisi olarak şu anki rolünde, eScience Center'ın yazılım sürdürülebilirliği ve FAIR yazılım çabalarının çoğunda yer almaktadır.

SIRS TF'ye
ilgi Carlos'un temel ilgisi, Hollanda eScience Center'ın FAIR yazılımı, yazılım dizinleri, yazılım arşivleme, yazılım kalite yönergeleri vb. alanındaki çabalarını (potansiyel) Avrupalı ortakların ve EOSC'ninkilerle uyumlu hale getirmektir. Şimdiye

6.3.13. Elisabetta Ronchieri

Organizasyon(lar)
1951 yılında kurulan İtalyan Ulusal Nükleer Fizik Enstitüsü (INFN), İtalya'ya yayılmış 20 bölümü, 4 ulusal laboratuvarı (Laboratori Nazionali di Frascati, Laboratori Nazionali del Gran Sasso, Laboratori Nazionali di Legnaro ve Laboratori Nazionali) bulunan bir devlet araştırma kuruluşudur. del Sud), Bologna merkezli Bilgi Teknolojisinde Ulusal Araştırma ve Geliştirme Merkezi (CNAF) ve diğer 2 ulusal merkez (TIPFA, Trento ve Lasa, Milano merkezli). Misyonu, İtalya'da nükleer, parçacık ve yüksek enerjili fizik araştırmalarını teşvik etmek, koordine etmek ve finanse etmektir. INFN, kuruluşundan bu yana, dağıtık bilgi işlem ve yazılım uygulamalarının kendi gelişmiş ihtiyaçları için açık ICT yenilikçi çözümler geliştirmektedir. Grid ve Cloud teknolojilerinde olağanüstü bir mükemmellik uzmanlığına sahiptir, AB tarafından finanse edilen ve sonunda Avrupa Şebeke Altyapısının (EGI) gerçekleştirilmesine yol açan birçok büyük Projeye liderlik rolleriyle destek vermiş ve katılmıştır. INFN, ana uluslararası Araştırma ile iyi kurulmuş işbirliklerine sahiptir.

Bilim dünyası için BİT çözümlerinin geliştirilmesinde yer alan merkezler ve özellikle EGI_InSPIRE ve EGI_Engage olmak üzere FP7 programı aracılığıyla Avrupa Komisyonu tarafından finanse edilen birçok projenin birincil ortağı olmuştur. INFN şu anda EGI_Engage'e katılım için (INAF ve INGV'yi içeren) İtalyan JRU'suna liderlik ediyor. INFN, Horizon2020 AB Araştırma ve Yenilik Çerçeve Programı kapsamında EOSC-Hub projesinin üç sütunundan biri olan INDIGO-DataCloud projesine liderlik etmektedir. INDIGO-DataCloud, bilimsel toplulukları hedefleyen, birden fazla donanıma yerleştirilebilen ve hibrit (özel veya genel) e-altyapılar üzerinden sağlanan bir veri ve bilgi işlem platformu geliştirmiştir.

Kısa Özgeçmiş
Elisabetta Ronchieri (PhD) bilgisayar bilimi mühendisidir. 2001 yılından bu yana INFN CNAF'ta çalışmakta ve DataGrid, EGEE, ETICS ve EMI gibi çeşitli AB projelerinde yazılım bakımı ve yazılım kalitesi için çözümler tasarlama ve geliştirme çalışmalarına katılmaktadır. CNAF'ta Elisabetta, bilgi işlem altyapısının operasyonları ve Ar-Ge'sinde yer alan bir teknoloji uzmanı ve yazılım geliştirme ekibinin üyesidir. IEEE NSS/MIC gibi uluslararası konferansların organizasyonunda işbirliği yapmaktadır.

Son zamanlarda, ana araştırmaları, yazılım mühendisliği konularında ve veri merkezi yönetiminde Makine Öğrenimi tekniklerinin rolünü araştırmak, DEEP-Hybrid-DataClouda ve IOTwins gibi diğer AB projelerinde işbirliği yapmaktan oluşmaktadır. Ayrıca, klinik anlatının tanımlanması gibi karmaşık sorunları ele almak için bilgi ve veriye dayalı yöntemleri birleştirmekle ilgilenmektedir. Elisabetta, teknoloji transferi projesi için Bologna ortaokulları ve veri analizi çerçevesi için Bologna Üniversitesi ile işbirliği yapıyor.

SIRS TF
Elisabetta'nın enstitüsüne ilgi, Açık Bilim ve Açık Kaynak Yazılımın destekçisidir. Şahsen, başkalarının bulguları doğrulayabilmesi ve üzerine inşa edebilmesi için tekrarlanabilir bir araştırmadan yanadır. Bunun için hem veriler hem de kodlar araştırmacılar için erişilebilir olmalı ve aynı sonuçları elde etmek için kolayca yürütülebilir olmalıdır. Veri ve kodun kalitesi ne kadar yüksek olursa, araştırmanın güvenilirliği de o kadar yüksek olur. Bu TF, yazılıma bir araştırma analizi içinde uygun bir rol verebilir.

6.3.14. Sam Yates

Kuruluş(lar)
İsviçre Ulusal Süper Bilgi İşlem Merkezi (CSCS). CSCS, İsviçreli ve Avrupalı araştırmacılar ve CERN ve MeteoSwiss dahil ortak kuruluşlar için yüksek performanslı bilgi işlem sistemlerini yönetir ve bunlara erişim sağlar. Donanım yönetimi ve kullanıcı desteğine ek olarak, CSCS ayrıca HPC ortamları için bir dizi yazılım aracı ve simülatörün geliştirilmesiyle ilgilenmekte ve hem Avrupa'da hem de dünya çapında birçok yazılım ve altyapı geliştirme projesinde işbirliği yapmaktadır. Daha fazla bilgiyi http://cscs.ch/about adresinde bulabilirsiniz .

Kısa özgeçmiş
Sam Yates, CSCS'de bir yazılım mühendisidir ve öncelikle İnsan Beyni Projesinin bir parçası olarak HPC sistemleri için bir nöron simülasyon kütüphanesi olan Arbor'un geliştirilmesiyle uğraşmaktadır. Akademik geçmişi esas olarak saf matematiktir, ancak kariyerinin çoğunda hem endüstride hem de akademide, jeofizik, telekomünikasyon ve en son olarak hesaplamalı sinirbilim alanındaki uygulamalarla bilimsel yazılım geliştirme ile ilgilendi.

projeleri etkileyen ilk elden sorunlarla karşılaştı. Bunlar genellikle araştırmacıları tanıyan ve sağlayan sistemler ile araştırma bağlamında yazılım geliştirme uygulaması arasındaki uyumsuzluklardan kaynaklanır. Modern bilimin yazılım bileşenini desteklemek için olgun bir altyapının bu sorunları iyileştirmeye yardımcı olacağını umuyor ve Sam bunun gelişimini desteklemekle çok ilgileniyor.

6.3.15. Moritz Schubotz

Organizasyon(lar)
FIZ Karlsruhe – Leibniz Bilgi Altyapısı Enstitüsü, sürdürülebilir bir bilgi altyapısı için yöntemler, süreçler ve hizmetler araştırır, geliştirir ve işletir. Kuruluş, açık ve yasal olarak uyumlu platformlar aracılığıyla veri, bilgi ve bilgi, yazılım ve hizmetler sunar ve bunları aranabilir, erişilebilir, birlikte çalışabilir ve yeniden kullanılabilir hale getirir. FIZ Karlsruhe, bilimde ve inovasyonda değer yaratma sürecini her düzeyde destekler ve araştırma sorularının yanıtlanmasını ve yenilerinin sorulmasını sağlar. Bunu yaparken, organizasyon "İlerleyen Bilim" yol gösterici ilkesini takip eder. Bunu yapmak için çeşitli platformlar ve hizmetler geliştirilmiştir. Özellikle swMATH, Zuse-Institute Berlin ile ortak bir çabadır. swMATH, matematiksel yazılımlar için ücretsiz olarak erişilebilen, yenilikçi bir bilgi hizmetidir. swMATH yalnızca matematiksel yazılımlar hakkında kapsamlı bir bilgi veritabanına erişim sağlamakla kalmaz, aynı zamanda yazılım paketlerinin ilgili matematiksel yayınlarla sistematik olarak bağlantısını da içerir. Amaç, swMATH'de kayıtlı bir yazılıma atıfta bulunan tüm yayınların bir listesini sunmaktır. Özellikle Zentralblatt MATH (zbMATH) içerisinde yer alan tüm makalelere yer verilmiştir. Hem bir programın arka planını ve teknik ayrıntılarını açıklayan makaleler hem de araştırma için bir yazılım parçasının uygulandığı veya kullanıldığı yayınlar olabilir. Daha fazla bilgi Zentralblatt MATH (zbMATH) içinde yer alan tüm makaleler verilmiştir. Hem bir programın arka planını ve teknik ayrıntılarını açıklayan makaleler hem de araştırma için bir yazılım parçasının uygulandığı veya kullanıldığı yayınlar olabilir. Daha fazla bilgi Zentralblatt MATH (zbMATH) içinde yer alan tüm makaleler verilmiştir. Hem bir programın arka planını ve teknik ayrıntılarını açıklayan makaleler hem de araştırma için bir yazılım parçasının uygulandığı veya kullanıldığı yayınlar olabilir. Daha fazla bilgi https://swMATH.org

Kısa Özgeçmiş
Moritz Schubotz, FIZ Karlsruhe - Leibniz-Institute for Information Infrastructure, Almanya'da Matematik Bölümü'nde kıdemli araştırmacıdır. Teorik bir fizikçi ve bilgisayar bilimcisi olarak matematiksel bilgi alma tutkusunu takip ediyor. Bilim, teknoloji, mühendislik ve matematikten akademik literatüre en son bilgisayar bilimi teknolojisini bir buket uygular. André Greiner-Petter, Philipp Scharpf ve Felix Petersen, matematiksel ifadeleri insanlar ve bilgisayarlar için daha kullanışlı hale getirmek için bu özel ilgiyi ve araştırma yöntemlerini ve araçlarını paylaşıyorlar. Bir Wikimedia Açık Bilim Mentoru olarak Moritz, FAIR ilkelerine bağlıdır ve genel olarak Açık Bilimi savunur. Doktora ile birlikte öğrenciler Cornelius Ihle ve Dennis Trautwein, Blockchain Teknolojisinin Açık Bilim hareketini ilerletme potansiyelini araştırıyor. 2014'ten beri NIST'de (Ulusal Standartlar ve Teknoloji Enstitüsü, ABD) saha dışı işbirlikçisi ve 2017'den 2018'e kadar NII'de (Ulusal Enformatik Enstitüsü, Japonya) burslu olarak çalıştı. TU Berlin, Almanya'dan Bilgisayar Bilimleri alanında.

SIRS TF'ye ilgi
Moritz , SIRS TF'deki ortakların, araştırma kültüründe, matematikte ve ötesinde temel bir değişimin çekirdeği haline gelecek kritik kütle olmasını beklemektedir. FAIR yazılımı, bilimde daha etkili işbirliğine katkıda bulunur. TF, FAIR yazılımının organizasyonunun temellerini açıklar. Oradan, araştırmacıların kendi alanlarına özgü sorunlarına odaklanabilmeleri için bir altyapı gelişecek ve arşiv, referans, tanımlama ve kredi yazılım eserlerini organize etmeye yönelik çerçeveler, musluk suyunu çalıştırmak gibi arka planda sorunsuz bir şekilde işlev görecektir.

6.3.16. leonardo kandela

Organizasyon(lar)
İtalya Ulusal Araştırma Konseyi (Cnr), İtalya'daki en büyük kamu araştırma kurumudur ve Araştırma Bakanlığı'na bağlı olarak multidisipliner faaliyetler yürüten tek kurumdur. 18 Kasım 1923'te tüzel kişi olarak kurulan Cnr'nin misyonu, kendi Enstitülerinde araştırma yapmak, ulusal endüstriyel sistemin yenilikçiliğini ve rekabet gücünü teşvik etmek, ulusal araştırma sisteminin uluslararasılaşmasını teşvik etmek, gelişmekte olan kamu ve topluma teknolojiler ve çözümler sunmaktır. özel ihtiyaçlar, Hükümete ve diğer kamu kurumlarına tavsiyelerde bulunmak ve insan kaynaklarının niteliğine katkıda bulunmak. Cnr'ın araştırma dünyasında ana kaynak, becerileri, bağlılıkları ve fikirleriyle insanlar anlamına gelen mevcut bilgidir. Bu sermaye, yarısından fazlası araştırmacı ve teknoloji uzmanı olan 8.000'den fazla çalışandan oluşmaktadır. Yaklaşık 4.000 genç araştırmacı, Cnr'de kurumun öncelikli ilgi alanları dahilinde lisansüstü çalışmalar ve araştırma eğitimleriyle uğraşmaktadır. Araştırma ortaklarından da önemli bir katkı gelir: Cnr'nin araştırma faaliyetlerinde yer alan üniversitelerden veya özel firmalardan araştırmacılar.

Kısa Özgeçmiş
Leonardo Candela, İtalya Ulusal Araştırma Konseyi, Istituto di Scienza e Tecnologie dell'Informazione "A. Faed".

geçiyor. Araştırma faaliyeti, araştırma ve geliştirmeyi yakından bağlayarak geliştirilmiştir. Aslında, Dijital Kütüphaneler ve Veri Altyapıları geliştirmeye çağrılan AB tarafından finanse edilen birçok projede yer almıştır ve D4Science.org altyapısının Strateji ve Portföy Yöneticisidir.

SIRS TF'ye ilgi
Leonardo açık bilime hayrandır ve onu teşvik etmek ve uygulamak için enerji ve çaba harcıyor. "Yayınlamanın" (halka açık hale getirme) açık bilim hareketini karakterize eden temel işlevsellik olduğuna inanıyor ve böyle bir kavramın geliştirilmesine katkıda bulunuyor. Ayrıca açık bilim için sistemlerin ve çözümlerin "sistemler sistemi" paradigmasına göre inşa edilmesi gerektiğine inanıyor. Bu, onu dergilerin ve arşivlerin tamamlayıcı bakış açılarını ve tekliflerini göz önünde bulundurarak "veri yayıncılığı" yaklaşımlarını incelemeye yönlendirir. Araştırma yazılımı, onun "yayıncılıkla" ilgili ilgisini çeken bir başka önemli araştırma eserini temsil ediyor. SIRS görev gücü, onun konuyla ilgili anlayışını tartışmak ve açık bilim uygulamalarına göre araştırma yazılımı yönetimine ilişkin fikirleri ileriye taşımak için eşsiz bir fırsattır.

6.3.17. Jean-François Abramatic (Başkan WG Mimarlık)

Organizasyon(lar)
Avrupa Açık Bilim Bulutu (EOSC) Yürütme Kurulu, Avrupa Komisyonuna tavsiyelerde bulunur ve EOSC'nin geliştirilmesi için tavsiyelerde bulunur.

Kısa Özgeçmiş
Jean-François, Fransız Bilgisayar Bilimleri ve Uygulamalı Matematik Araştırma Enstitüsü Inria'da Fahri Kıdemli Bilim Adamıdır. 2019-2020 dönemi için EOSC İcra Kurulu üyesi ve EOSC WG Architecture Başkanıdır.

SIRS TF'ye İlgi
Araştırmayı Açık Bilim paradigmasına doğru geliştirmek için yayınların, verilerin ve yazılımların bulunabilir, erişilebilir ve yeniden kullanılabilir olması esastır. Yayınlara (Açık Erişim), verilere (FAIR yönergeleri ve Açık Veri) özel ilgi gösterilmiş olsa da, araştırma yazılımlarını bilim adamlarının kullanımına sunma çabaları henüz başlangıç aşamasındadır. Bu nedenle, bu çabaların mevcut durumunu değerlendirmek ve araştırma yazılımına yayınların ve verilerin yanında birinci sınıf konumunu verecek girişimlerin planlanması önemlidir.

6.3.18. Martin Fenner

Organizasyon(lar)
DataCite, araştırma verileri, araştırma yazılımları ve diğer araştırma çıktıları için kalıcı tanımlayıcılar (DOI'ler) sağlayan, kâr amacı gütmeyen lider bir küresel kuruluştur. Araştırma topluluğu içindeki kuruluşlar, tüm araştırma çıktılarına DOI atayabilmek için DataCite'a üye olarak katılırlar.

Kısa Özgeçmiş
Martin Fenner, 2015'ten beri DataCite Teknik Direktörüdür. 2012'den 2015'e kadar PLOS Makale Düzeyinde Metrikler projesinin teknik lideriydi. Martin, Free University of Berlin'den tıp diplomasına sahiptir ve Kurul onaylı bir tıbbi onkologdur. Force11 Software Citation Implementation WG ve RDA/FORCE11 Software Source Code Identification WG'ye eş başkanlık eder ve EOSC Architecture WG'nin bir üyesidir.

SIRS'e
ilgi TF Martin, özellikle araştırma yazılımları için kalıcı tanımlama ve standartlaştırılmış üst veri ihtiyaçlarının ve araştırma yazılımlarının, yayınların, araştırma verilerinin, araştırmacıların, araştırma kuruluşlarının ve finansmanın birbirine bağlanmasıyla ilgilenmektedir.

---

[1] https://www.softwareheritage.org/2016/09/01/google-code-content-now-safely-collected-in-software-heritage/
[2] https://www.softwareheritage.org/2016/07/21/gitorious-retrieved/
[3] https://www.softwareheritage.org/2020/04/23/rescuing-250000-endangered-mercurial-repositories/
[4] Önemli bir not, *aynı yazılımın* aynı zamanda bazı araştırmacılar için bir araç, bir başkasının araştırmasının sonucu ve bir üçüncüsünün çalışma nesnesi olabileceğidir.
[5] Örneğin, bkz . Research Data Alliance'ın FAIR4RS Çalışma Grubu , FORCE11, Research Software Alliance ve "FAIR hizmetleri ve yazılımı" ile ilgili FAIRsFAIR görev 2.4.
[6] Bkz. https://en.wikipedia.org/wiki/Institutional_repository
[7] Bkz. https://en.wikipedia.org/wiki/Disciplinary_repository
[8] Bkz. https://www.microsoft.com/en-us/research/blog/new-award-honors-distinguished-artifact/
[9] Bkz. https://ec.europa.eu/commission/presscorner/detail/en/STATEMENT_16_2967

[13] Bu konuda daha fazla bilgi için https://uneskoprsist.org PERSIST projesine ve Software Preservation Network'ün çeşitli projelerine bakın.

[14] Bkz. https://www.softwareheritage.org/2016/09/01/google-code-content-now-safely-collected-in-software-heritage/ ve https://www.softwareheritage.org/2016 /07/21/gitorious-geri alındı/

[15] Örneğin, yayın için veri oluşturmak için kullanılan bir yazılım paketine atıfta bulunan yayına (Liang ve Kai Yong, 2013) bakın (Liang ve KaiYong, 2013) .

[16] https://www.openaire.eu
[17] https://www.zbmath.org
[18] https://www.semanticscholar.org
[19] Bkz. https://www.w3.org/TR/cooluris/

[20] https://software-guidelines.readthedocs.io/en/latest/
[21] İşte son derece popüler küçük bir kaynak kodu parçasının çok iyi bilinen bir örneği https://arstechnica.com/information-technology/2016/03/rage-quit-coder-unpublished-17-lines-of-javascript -ve-internet'i kırdı/

[22] Bu gerçek, örneğin DOI'nin bir örneği olduğu Handle sisteminin belirtim belgesinde açıkça belirtilmiştir: " *Bir tanıtıcı ile adlandırdığı varlık arasındaki tek operasyonel bağlantı, Sap Sistemi içinde korunur. Bu, elbette, idari bakımın bir işlevi olan kalıcılığı garanti etmez.* (Güneş ve diğerleri, 2003)
[23] Burada, yazılım değerlendirmesini uygulayan tüm büyük CS konferanslarında, Artefakt Değerlendirme Komitesinin makaleleri gözden geçiren program komitesinden ayrı olduğunu belirtiyoruz. AEC, makalenin kabul edilmesinden sonra devreye girer ve yazılım yazarlarıyla açık bir şekilde etkileşime girebilir.
[24] Bkz. https://blog.zenodo.org/2020/03/10/dryad-and-zenodo-our-path-ahead/
[25] Yeni ufuklar açan fikir için https://www.artifact-eval.org/ adresine ve gerçek konferans listesi ve güncel bibliyografya için http://evaluate.inf.usi.ch/ adresine bakın . ACM benzer bir rozet şeması benimsemiştir, ancak değerlendirme için belirli bir süreci zorunlu kılmamaktadır, bkz. https://www.acm.org/publications/policies/artifact-review-badging
[26] Örneğin bkz. https://popl20.sigplan.org/track/POPL-2020-Artifact-Evaluation
[27] https://paperswithcode.com
[28] Bu aynı zamanda UNESCO tarafından Ekim 2020'de yayınlanan Açık Bilim için taslak tavsiyelerin bir parçasıdır, Bölüm **II**, nokta (iv) (DG UNESCO, 2020) .